JULY/AUGUST
Vol. 3 No. 5
$3.00 U.S. Funds

AMERICA'S LARGEST TIMEX SINCLAIR MAGAZINE

TIME DESIGN

MAGAZINE

Super Summer Issue

Something For Everyone!!

VARIETY SALES
325 West Jersey Street,#2D
Elizabeth, New Jersey 07202

Hardware and Software for the: Timex/Sinclair 2068 and The Sinclair QL (Quantum Leap)

GREAT NEW ITEMS TAKE A LOOK!

FONTS & ICONS FOR

PIXEL PRINT, THE 2068 DESKTOP PUBLISHER- 102 CLIP ART ICONS, 6 NEW FONTS- CALL OR WRITE FOR DETAILS! ($19.95 EA), PIXEL PRINT IS STILL ALSO $19.95 !!

As close as your mailbox or phone: (201) 527 - 0535

Free delivery service available to Manhattan (NY, NY) addresses...call for details!

TRUMP CARD!!

* 896K RAM on your QL!
* TONY TEBBY TOOLKIT II!
* Print Spooler
* 2 RAMDisks (Dynamic & Static!)
* Dual Disk Interface!

All on one compact card!!

>>> & ALL FOR ONLY $299 <<<

QL REPLACEMENT KEYBOARD

The SCHON FULL-TRAVEL Keyboard is available NOW for your QL...

Installs EASILY- NO SOLDERING NECESSARY- just remove your QL top and replace with the SCHON top!! Retains the look and size of your QL- but with a REAL typewriter keyboard! > only $89 <

TOTALLY SPELLBOUND!

* 30,000 word spelling CORRECTOR FOR THE QL
* Load WITH your word processor for INSTANT spelling correction
* Several methods available for error notification- highlight, beep, etc.!

Add your own words- your QL expansion is the limit! For serious word processors- $49

>Front Page EXTRA<

All features of FRONT PAGE, the QL desktop publisher and MORE! Allows loading of text files from QUILL, the EDITOR, and almost ANY other file! FP extra AUTOMATICALLY MICRO-JUSTIFIES the text! Also, 5 new type-styles. Type styles can be changed WHILE text is loading...text loads into columns too!! More Features- Call or Write. only $49!!

For QL's with EXTRA RAM ONLY!

LIBERATE YOUR QL!

The DEFINITIVE BASIC compiler! Compiles almost ANY SuperBASIC program with no need for changes! As easy as typing LIBERATE! It will generate MULTITASKING code- Can even turn your programs into RESIDENT procedures (add commands to BASIC!) QLIBERATOR Easy to use, MANY more features- call or write! Or order for only $85.00

FUTURA ENVIRONMENT ON QL- QRAM!!

Windowing pointer utility system. Non-destructive windows, single key pop-up menus, FULL MULTITASKING, Super fast RAM disk,dynamic printer buffer, multi-mode screen dumps, and much more! Call or write for more info! QRAM is the environment of the FUTURA...and REQUIRES AT LEAST 256K RAM.<<

QRAM IS ONLY $49

64 COL. TERMINAL PROG. FOR YOUR 2068- NO EXTRA HARDWARE!

SPECTERM-64 is a 64 column terminal communication program for the 2068, and also has XMODEM error checking file transfers!! SPECTERM can also work with HARDWARE RS232 cards to run at 1200 baud! Works at 300 baud with the 2050 modem. EASY to use, with full documentation. 35K+ buffer area, 7K BASIC menu area...$34.95 pp. More features...call or write!

2068 DISK SYSTEM- $115!

CLARKEN DISK INTERFACE & DOS CARTRIDGE- WITH EXTENDED BASIC COMMANDS! VERY EASY TO USE, I.E.- PRINT #4:SAVE "FILE.BBB"...DOS TAKES NO RAM..NMI SNAPSHOT SAVE BUTTON...FULLY 2068/SPECTRUM COMPATIBLE...SUPPORTS 2 DRIVES, UP TO 360K/DISK. INCLUDES INTERFACE & DOS CARTRIDGE. DISK DRIVES ARE AVAILABLE SEPERATELY- CALL OR WRITE FOR DEAL!

FREE MONEY!! 1987 $ CATALOG AVAILABLE- $ SEND $1 AND GET $2* $ CREDIT! FREE MONEY! $

SEND ALL ORDERS TO:>>> PLEASE ADD $3 SHIPPING UNLESS OTHERWISE SPECIFIED.

VARIETY SALES
DEPARTMENT TDN-5
325 W. JERSEY ST.,#2D
ELIZABETH, NJ 07202

*LIMIT: ONE (1) $2.00 CREDIT PER PERSON.

JULY/AUGUST **Vol. 3 No. 5**

TIME DESIGNS MAGAZINE

FOR ALL TIMEX AND SINCLAIR COMPUTERS

TIME DESIGNS MAGAZINE CO.
29722 Hult Rd. • Colton, Oregon 97017

(503) 824-2658
CompuServe ID# 71350,3230

TIME DESIGNS MAGAZINE is published bi-monthly and is Copyright ©1987 by the **Time Designs Magazine Company,** Colton, Oregon 97017. All rights reserved.

Editor: Tim Woods

Assistant Editor: Stephanie Woods

Editorial Assistant/Production: D.L. Woods

Photography:
(unless otherwise noted): Thomas Judd
Printing by: Toad'l Litho Printing and Comp., Oregon City, Oregon 97045

SUBSCRIPTIONS: $16.95 a year for six issues (US funds only). No extra charge to Canadian subscribers. All other countries please write for information on air mail rates.

CUSTOMER SERVICE: Customer satisfaction is our goal. For subscription service problems please write or call TIME DESIGNS.

CHANGE OF ADDRESS: Write or call to prevent delay of sevice,

Next Issue:

Desktop Publishing

Renew Your Subscription Today!

EDITOR'S FORUM

It's very hard to think about doing anything constructive with your computer during the summer when so many other things confront you that are much more appealing. But never-the-less, I usually use this time of the year to work on one particular project. Something to keep the fingers limber--the brain from going stale, anything to prevent having to re-learn even the simplest procedures by the time "indoor" weather comes around again.

On this thought, I've compiled a list of some suggested summer computer projects:

1. "Key-in" (and de-bug) one of those extra long program listings that have appeared in TDM or other publications.

2. Make one of Tim Stoddard's modifications to your computer, or build a kit. (Note: If you don't have experience in this area, get help from someone who does.)

3. Buy your computer a present. How about a disk drive system or new 80 col. printer? Familiarize yourself with the DOS (Disk Operating System), or all of the features of the printer.

4. Tackle Syd Wyncoop's series on Z80 Machine Code in TDM (the first lesson is in the March/April '86 issue).

5. Explore a field of interest you are not familiar with, such as word processing, another computer language (FORTH, PASCAL, C, etc.), computer graphics, artificial intelligence, and telecommunications.

6. Write a program.

If your feeling very ambitious this Summer, you could also join a computer club, start your own Sinclair user group, revitalize the one your already a member of, start a BBS, and even organize a local Sinclair computer fair.

I would be interested to hear what you came up with this summer. If you have a tip or short program to share with others, send it in, and I'll print as many as possible in an upcoming issue.

A lot of news items have been filtering across my desk lately, and I'll try to keep you posted in the "For Your Sinclair" section. The TS world does not stand still. Many of you already know that the fine British magazine, ZX COMPUTING MONTHLY, is no longer being published. We here at TDM, feel this has created a real void for "serious" Sinclair computer users in the U.K. Plans are being carried out presently to unite what we are doing over here, with this international community and perhaps even entice the major hardware and software developers to take part.

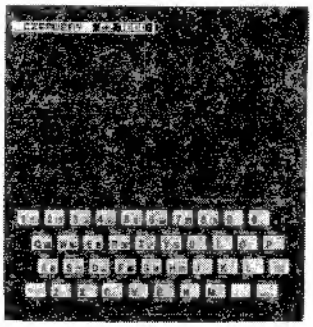
Another problem that has erroded over the past couple of months, is the lack of a reputable repair service for our Timex Sinclair computers. I am personally looking into this matter to see if it could be rectified. If you know of anyone who even repairs these machines as a hobby, please drop me a line.

Hope you enjoy the JULY/AUGUST issue. Find some shade, a hammock, a tall glass of ice tea, and give it a good read. We'll be back next issue.

Tim Woods
Managing Editor
TIME DESIGNS MAGAZINE

For Your Sinclair

TS Machines Thrive In South America



When the Timex Computer Corporation here in the States closed their doors for good, a rumor was going around that thousands of TS2068's had been "dumped" upon the South American consumer electronics market. But the rumor couldn't be substantiated and was soon forgotten. Later on, an "unauthorized" Spectrum clone was discovered in Brazil, along with a dubious software supply. The manufacturer of the Brazilian Spectrum had even improved some of the bugs in the original Sinclair ROM. Versions of this improved ROM found it's way here in the U.S., and some Spectrum Emulator boards for the TS2068 featured the chip.

The extent of the Timex Sinclair market in South America wasn't realized until just recently, and it appears that the TS activity is in Argentina, the most educated country in Latin America (and ranks very high world-wide). Christian Pusso, Director of the largest computer publication in Argentina, called "K64", recently reported to TIME DESIGNS, "The market here in Argentina is strongly formed by Commodore 64, 128, MSX compatibles, Timex Sinclair, Atari 800/130, Apple Macintosh, IBM PC's and its clones...all assumed to be a total of 500,000 home and PC users".

Laws are very strict in some of the South American countries on importing large quantities of computer goods, forcing many manufacturers (including Commodore) to license existing companies within the country to assemble the computers and then distribute them. In this case, the CZERWENY (an odd-sounding name) Company is licensed through Timex (and more than likely, Timex of Portugal) to manufacture the CZ1000 (the TS1000) and the CZ2000 (the TS2068). Other Timex "clones" and Spectrum clones are widely available...along with stock (U.S. versions) of the Timex Sinclair line.

Many South Americans are discovering the powerful capabilities of the TS machines, and their lower cost is especially appealing. Many computer dealers feature add-ons and software. Devices to convert the 2068 into a

Spectrum, like emulators and twister boards are the most popular. Spectrum games and other programs are readily available.

The impact of this South American Timex Sinclair "connection" may be far-reaching. The need for an information exchange is great, and currently there is a great interest in programming and low-cost disk drive systems.

Sinclair Clones In The Kremlin

TDM contributor, J. Kevin Paulsen, brought this to our attention: In the March 16th issue of INSIGHT Magazine (published by The Washington Times) an article on the Soviet Union's attempt at catching up with the microcomputer technology race was featured. The interesting article pointed out how the Soviet's have surpassed the West in many areas except for the development of inexpensive personal computers for the masses. The author reports that, "It is a system in which access even to typewriters and copying machines is carefully guarded, out of fear of dissident and other unofficial communications. Printers are not available on the open market, and computer components are prohibitively priced".

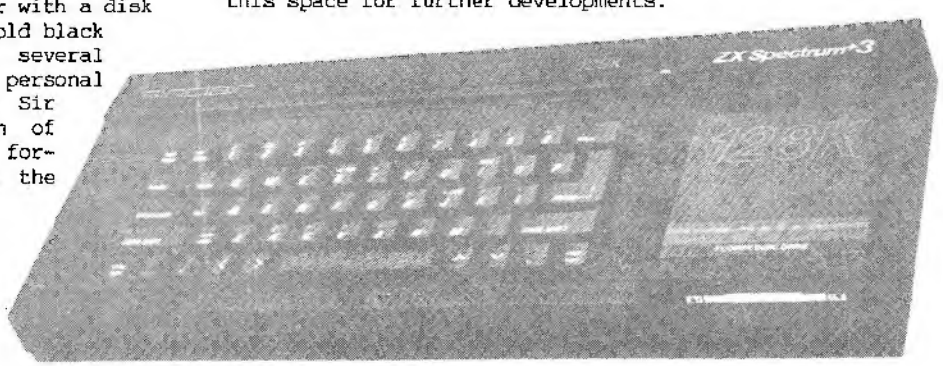
But the most surprising revelation is that the Soviets have started to manufacture computer "clones". "Smaller computers with brand names such as Agat, Iskra, and Electronika began to dribble off assembly lines, closely resembling machines from IBM, Apple Computer Inc and Sinclair Research Ltd.".

Newest Sinclair

TDM correspondent, Bob Lussier, supplied us with information on the new SPECTRUM PLUS 3 computer which will replace the current 128K+2 model shortly. The computer makes history as the first Sinclair with a disk drive unit, and it is also a return to the old black styling we had grown accustomed to, plus several other new features makes this the ideal personal computer. It's sad though to think that Sir Clive had no hand in this latest version of the time-tested ZX Spectrum. Instead, the forward marketing strategies of Amstrad are the driving forces behind this new micro.

Along with the built-in 3" Amstrad floppy drive, there is also a built in Centronics printer port, an auxiliary disk drive port for adding a second drive, MIDI port, RGB monitor connection, twin joystick ports, and standard cassette tape I/O ports. The DOS is a version of Amstrad's own Amsdos, but has been modified, and uses the old Microdrive commands exclusively. The price of the Spectrum Plus 3 is £249, and includes a disk with six games, plus a joystick. The Spectrum Plus 2 now sells for £149. For further info, you can write to: Sinclair, Brentwood House, 169 Kings Road, Brentwood, Essex CM14 4EF, U.K.

CP/M is compatible with the built-in disk operating system, but Amstrad has released no plans to offer a CP/M emulator. However, outside developers are currently working on such a system. In fact, most of the large software houses in Great Britain are enthused about the new Spectrum and are gearing up for production of new game software supplied on the 3" disk format. To quote one source, "From a software point of view the 128K Plus 3 means that huge games--similar to the best of the US Commodore 64 disk-based programs will be possible. The way these work is to load in sections of the game as you play--deleting those areas of the game you've already played through from memory, as you go".



More Computer Fests Announced

"Mini-Fair" Comes to the Northwest

Plans abound for regional and national TS computer fests. It has been said that the future of our Timex Sinclair community depends on gatherings such as these. Now more users will be able to attend this next year due to a variety of locations currently being proposed.

The three user groups from the Bay Area of Northern California are currently making plans for a "Silicon Valley TS Computer Fest" to be held somewhere in the South Bay Area, and a suggested date of June 1988. The region offers itself as an excellent family vacation spot, as well as the "bargain hunter's dream"...several large electronic parts houses are nearby. For further information write to: Norm Lehfeldt, 757 Guerrero, San Francisco, CA 94110.

Four user groups from Florida are currently working on details for the "Sunstate TS Winterfest '88" to be held in Orlando, Florida, either at the end of February or first part of March 1988...an exact date has not been set yet. The Central Florida area is the number one vacation destination in America, with Walt Disney World nearby. Along with U.S. dealers of Sinclair merchandise, the planning committee will invite companies from the U.K. as well. A 24 hr. BBS has been set up to serve as an information exchange on the Winterfest. The phone number is: (904) 775-0093 (7/1/N). Or write to: Mary-Lynn Johnson, 249 N. Harden Ave., Orange City, FL 32763.

Good News for TS users in the Northwest--the "Timex Sinclair Mini-Fair" will be held on Saturday, September 26th, 1987, in Seattle, Washington. Hours are 10 am to 5:30 pm, and small admission charge of $2 per person ($3 for family) will be charged at the door. Five user groups from the region will be participate, along with at least six TS dealers. The TS Mini-Fair is sponsored in part by Time Designs Magazine and RMG Enterprises. Nine guest speakers are planned for a wide variety of Sinclair topics and interests. As of this writing the exact location had not been established. For further information write to: TDM, 29722 Hult Rd., Colton, OR 97017.

Other TS computer fests have been discussed for both the Midwest again, and one for the Eastcoast. Watch this space for further developments.

NEW PRODUCTS ANNOUNCEMENT
For TS2068 and SPECTRUM

NovelSoft

brings Artificial Intelligence
to the Timex/Sinclair

Introducing...

ZXPERT

ZXpert lets you explore the reasoning process of human experts.

With ZXpert you can create an Expert System on any topic you like!

- * A useful and educational program
- * Easy and fun to use
- * No programming language to learn
- * On-line help
- * Includes 2068 & Spectrum versions
- * Supports Microdrives
- * Two example knowledge bases
- * Superb Manual with Tutorial

plus...

Once you have developed a useful knowledge base, with your permission NovelSoft will sell it and pay you a royalty!!!

Join the revolution with...ZXpert

$19.95 U.S. Plus $3.00 S & H

NovelSoft
A FORMAT FOR THE FUTURE

33 Skelton Street, Toronto, Ontario, Canada M8V 3W3 * TEL. (416)259-8682 * CompuServe 70416,1435

Product/Dealer News

Larry Kenny of LARKEN ELECTRONICS told TDM about the plans for the 256K RAMdisk for the TS2068. It will use static-type RAMs complete with battery back-up. It will require users to purchase the Larken cartridge, as it contains the commands for the RAMdisk. Larry strongly urges any 2068 user who is interested in increasing their memory up to 256K to write to him, as he is compiling an "inform when ready list", plus it will give Larry an idea of how much interest there is in this product. Also available now, are two new versions of the Larken EXTENDED BASIC/DOS CARTRIDGE, one for users of the Aerco FD-68, and another for Ramex Millenia K/SPDOS Disk System owners. Both versions feature full Spectrum compatability and NMI (non-maskable interrupt), plus improves the original DOS. For further details and prices, write to: Larken Electronics, RR#2, Navan, Ont., Canada, K4B-1H9.

WEYMIL CORPORATION (Box 5904, Bellingham, WA 98227) has released the DELTA DEVICE, a non-volatile memory system for the TS1000. A small circuit board contains 32K RAM divided into four 8K blocks which can be independantly switched via a DIP switch to various locations in the Sinclair memory map. Memory is backed-up by a battery. Included is the Rigter Operating System (ROS), which is a data handler/directory system manipulated by simple commands. Application of the Delta Device includes the ability to store several programs and recall them instantly. Price: $75.00 plus $5 (S&H).

In last issue's product/dealer news column, we reported on the excellent expanded new version of THE GUIDE TO TS TELECOMMUNICATIONS, but we incorrectly stated the price as $5...the correct price is $7.50, and worth every penny. The book is strongly recommended to anyone wanting to learn how to use modems with their Sinclair computer. Lots of information for the advanced user too. Send for your copy: Pete Fischer, P.O. Box 2002, Tempe, AZ 85281.

Bill Jones (author of SMART TEXT) is starting a newsletter called: "TS-2068 Safe Disk Up-Date". Emphasis will be on programming, as well as the Oliger Disk Drive System. It will be published quarterly, and the annual subscription price of $12.00 includes unique dividers and punched pages to fit a 3-ring binder. For further information write to: TS-2068 Safe Disk Up-date, 1317 Stratford Ave., Panama City, FL 32404.

Silicon Mountain Computers (C-12, Mtn. Stn. Group Box, Nelson, B.C., Canada V1L 5P1) has released an excellent new modem terminal program for the TS1000 called ZX-TERM*80. The software package uses genuine hi-res graphics to provide up to 80 columns of text (!) and windowing. Features include XMODEM protocol for uploads and downloads, printer support, Upper/Lower case characters, Westridge or Byte-Back compatible, and much more. Price: $24.95. Write for complete details.

Robert Hartung reviewed the Spectrum program, BETA BASIC in the MAY/JUNE '87 issue of TDM. A new version (4.0) is available for Spectrum's with 128K. Price is £15.95 and is available from: Betasoft, 92 Oxford Rd., Moseley, Birmingham, B13-9SQ, U.K.

Markel Enterprises has a new address: 4712 Avenue "N", Suite 383, Brooklyn, NY 11234; and also a new BBS for Timex Sinclair users, called SINCLAIR AT NIGHT (718) 627-1293 (settings are 8/1/N).

RMG Enterprises has a Timex Sinclair BBS—(503) 656-8072, hours from 9 pm to 10 am (settings are 8/1/N).

Variety Sales (325 West Jersey St., #2D, Elizabeth, NJ 07202) is giving a free British Sinclair magazine (while supplies last) with each order received and the customer mentions they saw mention of Variety Sales in TDM.

Sir Clive's new battery powered laptop computer, the Z88 has had some delays in delivery due to "bugs" in the software. All problems should be corrected by the time you read this. There is a limited special offer price for the Z88--only £229.95. Write to: CAMBRIDGE COMPUTERS LTD., Cambridge, CB4 1BR, U.K.

Sunset Electronics (and some selected dealers) have new dust covers for your Timex Sinclair 2068, 1000, 1500, 2040 printer, and the QL. Made of quality fabric, these attractive covers prevent damaging liquid spills, and harmful dust. The logo tells everyone your proud to own a Sinclair. Write to Sunset for complete price list: 2254 Taraval St., San Francisco, CA 94116. Telephone-(415) 665-8330.

Users Group Update

Have we listed your group recently?
Send us the club's name/address and get noticed!

Southeastern Michigan Computer Organization
Timex Sinclair Special Interest Group
c/o Barry Carter, newsletter editor
Box 614
Warren, Michigan 48090

GUTS/SV (Group Using TS of Silicon Valley)
c/o Bill Miller
6675 Clifford Drive
Cupertino, California 95014

Clackamas County Area TS Users
c/o Rod Gowen
1419 1/2 7th Street
Oregon City, Oregon 97045

Kansas Area Timex/Sinclair Users Group
c/o Paul Reynolds
4557 Cherry
Wichita, Kansas 67217

Vashon Island Sinclair Timex Association
c/o Tony Willing
P.O. Box 199
Vashon, Washington 98070

Seattle Area Timex User Group
c/o John Scearce
P.O. Box 88361
Seattle, Washington 98188

Tri-City Timex Sinclair User's Group
c/o Mike Davis, President
706 S. Mason
Saginaw, Michigan 48602

TAS-BAM User's Group
P.O. Box 48961
St. Petersburg, Florida 33743

Sun Lake User Group
Bill Ward, Secretary
1200 Lake Drive
Grand Island, Florida 32735

Timex Sinclair User Group-Philadelphia
P.O. Box 53490
Philadelphia, Pennsylvania 19105

Westmoreland Computer Users Club
Timex Sinclair Special Interest Group
P.O. Box 3051
Greensburg, Pennsylvania 15601

WMJ DATA SYSTEMS

Serving the Timex/Sinclair community since 1983

- Free shipping on all prepaid orders!
- Shipment within 24 hours on most of our orders!
- Your personal check is gladly accepted with no delay!
- C.O.D. phone orders accepted! (Shipping charges added)

TS1000

ACZ GENERAL LEDGER	TS1000	$19.95
SUPERTAPE-45 PROGRAMS	TS1000	18.95
TEXTWRITER 1000-32 col.	TS1000	12.95
SPEECH SYNTHESIZER	TS1000	16.95
STOCK PLOT	TS1000	14.00

BOOKS AND ACCESSORIES

KRAFT SYSTEM JOYSTICKS	TS2068	$ 9.95
HACKER'S HANDBOOK	ALL	9.95
BASICS AND BEYOND	TS2068	7.00
C10 CASSETTES	25 for	21.00
C20 CASSETTES	25 for	23.00

TS2068

ACZ GENERAL LEDGER	TS2068	$19.95
THE DEALER'S DEN	TS2068	19.95
MACHINE CODE TUTOR	TS2068	18.50
TOMAHAWK	TS2068	16.95
NIGHT GUNNER	TS2068	16.95
FIGHTER PILOT	TS2068	16.95
CRITICAL MASS	TS2068	15.95
SABOTEUR	TS2068	15.95
BASIC TOOLKIT	TS2068	13.95
MSCRIPT (specify conf.)	TS2068	24.95
DIAMOND MIKE	TS2068	16.95
COLONIZE THE UNIVERSE	TS2068	16.95
STOCK PLOT	TS2068	16.00
All QUICKSILVA Software	TS2068	CALL

WMJ DATA SYSTEMS—TD

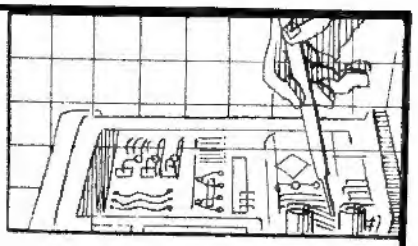
4 Butterfly Drive
Hauppauge, NY 11788

(516) 543-5252 (2 p.m. - 5 p.m. Best)

FREE Apple, Atari, Commodore, IBM...Catalog
FREE Timex/Sinclair Catalog - Write or Give us a CALL!

TS Communique

By Joe Williamson



A forum for people having problems with their 1000, 1500 and 2068. If you would like to ask a question, send it to:

TS Communique
c/o Time Designs Magazine Co.
29722 Hult Road
Colton, OR 97017

I bought a QL and would like to know how to hook it up with my Zenith Data Systems RGB Monitor. If you can supply any help, I would be very grateful. Thank you.

Larry Anderson
Davenport, IA

Dear Larry,

From the pin-out you sent for the Zenith monitor, the following connections should work. Be careful in soldering so that there arn't any frayed wires or solder bridges causeing shorts between pins.

QL	Signal	Zenith Monitor
2	Ground	16 & 19
3	composite video	20
4	composite sync	1
5	vertical sync	14
6	Green	5
7	Red	7
8	Blue	6

If this does not lock the picture into sync, try connecting pin 4 of the QL to pin 1 & 14 and eliminating the pin 5 connection of the QL. Also, I showed the connection of composite video so you can display either RGB or monochrome video which may help with certain video modes. -Joe

I am using a Timex 2068 computer, driving a brand-new Magnavox monochrome monitor (Model BM 7622), working mainly with MSCRIPT word processing software. I continually notice the following symptons:

1. Usually, the malfunction first appears as a flickering of the screen display. After about ten minutes, the video signal is lost and the screen shows only a normal raster (blank lit screen). Sometimes the signal is not entirely lost; very dim words and letters of the text are visible on the screen.

2. Manipulating the brightness and contrast controls of the monitor does not restore the video signal to normal. Turning the monitor power off then on again does not restore the display.

3. Turning the 2068 off then on generally restores the signal to normal, but then of course everything in memory is lost.

4. The remedy that seems to work most of the time is to pull out the plug from the 2068 monitor jack and then plug it in again, whereupon the video signal usually returns to normal strength.

5. The symptons described above are the same whether I am using a three-year-old 2068 or a two-month-old 2068, a fact which seems to indicate that the Magnavox monitor is at fault. However, there still could be something about the 2068 monitor output circuit which is causing the problem.

6. The output from the RF jack of the 2068, which feeds an ordinary television set, remains constant and strong without any of the symptons described above.

Seymour Miller
Forest Hills, NY

Dear Seymour,

Because you have the problem with both computers, It would be most reasonable to think that there is a problem with the monitor or the monitor cable. Try jiggling the cable at both ends to see if the symptons come and go or replace the cable. To see if the problem lies in the computer, try another monitor. If there is a problem with the computer, there are several places to check. Video is processed in the SCLD and comes out on pin 33 and on the RGB lines (pins 48, 47, & 46). You would need an oscilloscope to check for activity on these pins.

The video then goes through the sync & white level control, VR1 (try adjusting it to see if this clears up the problem) and feeds the base of Q5. the signal comes out on the emitter and feeds the emitter of Q4 and out the collector of Q4. From here, the video feeds both the RF modulator and the video output circuit consisting of Q3 and Q4 which have been known to cause problems. The video signal feeds the base of Q3, out from the collector, feeds the base of Q2 and out on the emitter through C58 to the video out put jack. Determining where the video is found and then lost can tell you where the fault is quite easily (one of the easier circuits to follow). I have seen alot of video problems traced to the SCLD (unfortunately) and Q2 & Q3.

As I stated before, your problem sounds like it is in the monitor or cable. Look there first! -Joe

I read the letter from Mel Routt in the first TS Communique. I seem to have the exact same problem with MSCRIPT. I have an Aerco parallel printer interface connected to a Blue Chip M120/10 printer. I get the random printout glitches on both tape and disk versions of MSCRIPT. Tasword II and anything typed from Aerco RP/M work fine. I am also looking for a CP/M word Processor.

For some reason the print from my Alphacom 32 is smaller than usual. The letters vary in size during the printout. This happens on both the 2068 and the 1500 so the problem is in the printer. Is there a solution for this?

I have an Aerco FD-68 Disk interface connected to a Zenith ZVM-131 RGB monitor. I can see faint lines moving on my video. In RP/M, the letters appear to shimmer. This appears to be some sort of interference. The cable from the FD-68 to the monitor is shielded so the interference does no come from there. Someone told me that it may be coming from the 2068 power supply. Is a better power supply available?

Dave Bennett
Lemoyne, PA

Dear Dave,

Mel Routt wrote us back and said that since he received Version 5.2 of MSCRIPT, the problem has disappeared. So try trading up for the new version. The Alphacom 32 problem sounds like a damaged pin heater. Your best bet is to replace the whole unit. The interference problem sounds like a buss loading problem or poor sync. If the problem is there with only the disk interface connected, more than likely it is from poor sync being stripped from the video. Try adjusting the sync and White level adjustment inside the computer just to the top upper left of the speaker designated as VR1. If this doesn't help, you may need to check the actual sync stripping circuit on the disk interface. If you have an oscilloscope, look at the sync line. You should have clean, steady negative peaks with no tearing where the peak starts or ends. -Joe

In your Banner progran for the 2040 printer (TS 1000) in SUM Magazine, May 1986, I can't get past the line:

```
30 LET LEN=LEN M$
```

It comes up with a syntax error. I have tried several things such as "" etc. to no avail.

A. L. Francis
Yucaipa, CA

Dear Mr. Francis,

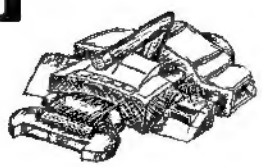
LET and the 2nd LEN are both key words and need to be entered as a single keystroke. The first LEN is a variable and is just typed in L-E-N. -Joe

STARFLEET 2068

SPACE ARCADE GAME

by

STAN LEMKE



STARFLEET 2068 is an all BASIC, low-resolution graphics, shoot 'em up space wars game for the Timex Sinclair 2068. The program listing and CK-TYPE output (to aide your typing/debugging of this program--see May/June '87 issue of TDM) is printed below. STARFLEET 2068 is also available on tape from Lemke Software (2144 White Oak, Wichita, KS 67207) for $9.95 ppd., in case you want to try this game, but don't have the time to key it in.

Scenerio: In the late 1990's, NASA finally perfected the first WARP DRIVE engine. Earth Federation, a joint US-Canada-European space agency quickly arose to control deep space exploration. Over the first several decades, 9 major STAR BASES were constructed; 1 at the edge of our solar system, and 8 others nearly equidistant from this home base. The year is now 2068 AD. After nearly 70 years of exploration, it was beginning to look like we were alone in the galaxy...until THEY appeared. They made no effort to answer our communications, gave no sign of peaceful intent--and as far as we know, might be purely mechanical (computerized) devices. They simply came, destroying everything in their paths. They caught us by surprise. After 70 years of uninterrupted peace in space, Earth Federation had only 1 armed Star Ship left in space active service--the STRATA-GEM. The ship was hastily fueled, its crew briefed, and as the aliens were about to attack Star Bases 1 and 8, the STRATA-GEM was launched to meet the attackers. YOU are the CAPTAIN. Battle Stations are sounding. Good Luck!

Armament: The Strata-Gem has Electro-Magnet "shields" to repel the enemy weapons. Full strength is 10. If they go to zero, good bye!

Weapons: You have Photon Torpedos (for longer range shots) and Phasor weapons (for closer range shots). The number of each depends on the level of difficulty.

Sensors: Sensor range is 3 parsecs, at full strength, but will vary on shield strength (ship damage).

Engines: You have WARP Speed capability (Warp 1 to 3). You have 20 units of fuel. (Note: the faster you go, the more fuel you use!!) WARP 1 moves you 1 parsec (row and/or column) per move, WARP 2 moves you 2 rows and/or columns per move, etc. If you run out of fuel, good bye!

Docking: You must DOCK to refuel, repair, restock armament, etc. The Star Base will not lower its shields (necessary for you to dock) if there are enemy ships in the quadrant.

Surprises: Yes, there are a few. If a Base refuses to lower it's shields, try-try again. You may have just been a "Bad Captain" and need to pay penance! There are 6 levels of difficulty!

How to Play.

STARFLEET 2068 has 2 screens, the navigation screen and the weapons/tactical screen. The navigation screen shows you a map of the galaxy: stars, bases, your ship, the enemy ships. The weapon/tactical screen shows you a "close-up" view around your ship, also the range of your sensors (a circle). Anything inside this circle is "fair game" for your weapons.

On the Navigation screen, "HELM AT YOUR CONTROL" will flash, telling you that it is your turn to make a move. Use the joystick to command your ship to make a move. You can change the WARP SPEED by pressing numbers 1 to 3. Press "D" to DOCK, press "W" to go to the WEAPONS screen. NOTE: to DOCK, you must "fly" your ship into the STAR BASE User Defined Graphic you wish to dock with. Then on the next turn you command "D" (dock). DON'T try to fly thru stars or other ships! Press "0" (zero) for no move, as in station keeping.

On the Weapons screen, you will see the map "close-up". Here you will see that Docking really places you just beside a STAR BASE. "WORKING!" will flash when you first go to this screen. The program is calculating the range, distance, position of everything for this new display. When the screen is completed, you will see your ship (center of the circle) with a " " User Defined Graphic flashing on it. This is your weapons "cross-hairs" for aiming at a specific target. Use your joystick to position this on the target of your choice. When ready, press fire. At any point prior to pressing fire, you can select your weapon: press "T" for torpedo, "P" for phaser weapon. If you shoot at yourself, this is a signal to abort the shot. Press "N" to go to the Navigation screen.

The program (listing).

Type in the program, SAVE it to tape with RUN 4040 [ENTER], (you start the program with RUN [ENTER]). LOAD in the CK-TYPE program (see May/June '87 issue of TDM) and run it to help you find typing errors. Correct these as required, and re-SAVE the new copy.

This program uses 13 User Defined Graphics as 8X8 pixel Sprites. These are easily identified in the program listing as single INVERSE VIDEO letters (A thru M). I did this by re-defining each UDG as the inverse of its letter. This has no effect on the CK-TYPE, and requires no action on your part except that each time you encounter an INVERSE VIDEO character, you know it is a User Defined Graphics character (and which UDG character) and you must type it in with the "GRAPHIC" cursor command.

The program also abounds with REM statements. These are quite helpful for programming/debugging purposes, but if you remove these from your "working" copy, you will increase the program speed by at least 25%!!

```
 10 REM  **********************
 20 REM   S T A R     F L E E T
                2068 AD
 30 REM  **********************
 40 REM © by S D Lemke      1984
              2144 White Oak
              Wichita, Ks. 67207
 50 REM  **********************
 60 DEF FN a(f)=INT (f/32)
 70 DEF FN b(c)=c-FN a(c)*32
 80 DEF FN c(f,c)=INT (f*32+c)
 90 DEF FN d(x)=FN b(x)*8-4
100 DEF FN e(x)=171-FN a(x)*8
110 DEF FN f(x)=INT ((176-x)/8)
120 DEF FN g(x)=INT (x/8)
130 DEF FN h()=INT (1PEEK 23674
*65536+PEEK 23673*256+PEEK 23672
)/60)
140 GO TO 3510: REM [illegible]
150 REM
160 REM *** Clear Messages ***
170 REM
180 PRINT AT 19,0;TAB 31;" ";TA
B 31;" ";TAB 31;" ": RETURN
190 PRINT AT 17,0;TAB 31;" ";TA
B 31;" ";TAB 31;" ";TAB 31;" ";T
AB 31;" ": RETURN
200 PRINT AT 19,0;TAB 31;" ";TA
B 31;" ";TAB 31;" ": PRINT #0;AT
 0,0;TAB 31;" ";TAB 31;" ": RETU
RN
210 PRINT AT 0,10;"Time = ";FN
h(): RETURN
220 PRINT AT 15,10;"Time = ";FN
 h();TAB 31;" ": RETURN
```

STARFLEET 2068

```
 230 SOUND 0,t 1 ,1,t 2  2,t 3
3,t(4 ,4,t(5  5 t 6),6 t 7  7 t
8) 8 t(9),9,t(10 ,10,t 11 ,11,t(
12 ,12,t(13) 13 t 14   PAUSE 15
 SOUND 8,0,9,0,10 0  RETURN
 240 REM
 250 REM *** Fuel Status ***
 260 REM
 270 LET i=INT (fuel+.99)   IF fu
el<0 THEN CLS   PRINT "*********
*********************** You have
 run out of fuel!        The Fede
ration is doomed.        Your Shi
p is lost.              *********
************************"  GO TO
3940
 280 FOR j=1 TO i  GO SUB 210  L
ET f$(j)="■   NEXT j  FOR j= i+1
) TO 20: LET f$(j)="     NEXT
PRINT AT 18,0,"

 290 IF i>=10 THEN PRINT AT 18,0
,"FUEL = ", INK 4,f$,TAB 28, INK
 9,fuel
 300 IF i>5 AND i<10 THEN PRINT
AT 18,0,"FUEL = ", INK 6,f$,TAB
28; INK 9;fuel
 310 IF i<=5 THEN PRINT AT 18,0,
"FUEL = ", INK 2,f$,TAB 28, INK
9,fuel
 320 REM
 330 REM *** Weapons Status ***
 340 REM
 350 PRINT #0,AT 0,0, Torpedo= '
,pt,TAB 18;"Phaser= ',ph,TAB 31
" "
 360 PRINT #0,AT 1,0;"Shields =
" INT (sh*10)/10;TAB 18 "Sensors
 = ";INT (sn*10)/10,TAB 31 ' "
RETURN
 370 REM
 380 REM
 390 FOR i=1 TO 10  GO SUB 210
LET b(i,1)=FN a(l(i))  LET b i,2
)=FN b(l(i))-1  NEXT i  FOR i=1
TO 9  LET c(i 1)=FN a( l(i+10))
LET c i,2)=FN b(l(i+10))+1  NEXT
 i  LET d 1)=FN a(l(31))  LET d(
2)=FN b(l(31))-1  LET c(21,1)=d(
1)  LET c(21,2 =d(2)
 400 PRINT AT 0 0, INK ink7,a$
 410 LET je=0  FOR i 1 TO 10  GO
 SUB 210  IF l(i)>0 AND l(i)<704
 THEN PRINT AT b(i,1) b i 2)  IN
K ink5,"■"  LET je=je+1  LET g(j
e)=i
 420 NEXT i
 430 FOR i=1 TO 9  GO SUB 210  I
F l(i+10)>0 AND l(i+10) 704 THEN
 PRINT AT c(i,1) c(i,2)  INK ink
1 "■
 440 NEXT i
 450 PRINT AT d 1) d 2   INK ink
0 ■
 460 GO TO 270
 470 REM
 480 REM *** Read Joystix ***
 490 REM
 500 OUT 245 14  LET stk=IN 5110
  IF stk<128 THEN LET b$=CHR$ 13
  LET stk=0  RETURN
 510 LET stk=255 stk  RETURN
 520 REM
 530 REM *** Target Lock ***
 540 REM
 550 LET dlx=(x2-x1)/20  LET dly
=(y2-y1)/20  IF ABS dlx =1 AND A
BS dly<=1 THEN RETURN
 560 PRINT AT 18,0,TAB 31," ' TA
B 6; FLASH 1,"Locking onto Targe
t!"; FLASH 0,TAB 31 "  "
 570 FOR k=5 TO 20  GO SUB 220
LET xp=x1+INT (dlx*k)  LET yp=y1
+INT (dly*k)  IF POINT (xp,yp) T
HEN GO TO 590
 580 NEXT k  GO SUB 180  RETURN
 590 IF ABS (x2-xp)<=4 AND ABS (
y2-yp)<=4 THEN GO SUB 180  RETUR
N
 600 LET x2=xp  LET y2=yp  LET r
x=FN f(y2  LET cx=FN g(x2)  GO
SUB 1 0  RETURN
 610 REM
 620 REM *** Nav Menu ***
 630 REM
 640 LET move=0  GO SUB 390  LET
 tim0=FN h
 650 GO SUB 180
 660 LET timer=FN h() +20-rank*2
 PRINT AT 21 0  FLASH 1  NAVIGAT
ION"  FLASH 0          WEAPONS
DOCK'
 670 GO SUB 210
 680 LET b$=INKEY$
 690 IF FN h()>timer THEN GO TO
1130
 700 IF b$= L  OR b$='l' THEN GO
 TO 1580
 710 IF b$="d' OR b$= D  THEN GO
 TO 1400
 720 IF b$  n  OR b$= N  THEN GO
 TO 750
 730 GO SUB 500  IF stk >0 THEN
GO TO 750
 740 IF b$ <CHR$ 13 THEN GO TO 6
70
 750 REM
 760 REM *** Navigation ***
 770 REM
 780 IF wf>mwf THEN LET wf=mwf
 790 GO SUB 180
 800 LET tim0=FN h    PRINT AT 2
0,0, FLASH 1       Helm at your
 command.         AT 19 8  FLASH 0
  Warp Factor =   wf  LET ist=0
 810 GO SUB 210  LET b$=INKEY$
 GO SUB 500  IF b$  0  OR stk  0
THEN LET ist=1  PRINT AT d 1  d(
2)  PAPER ink3   '  IF s(1)  0 A
ND s(2   0 THEN PRINT AT s(1),s(
2)  INK ink1 "■"  LET s(1)=0  LE
T s 2)=0
 820 IF b$>="1  AND b$ = 3  THEN
 LET wf=(CODE b$)-48  PRINT AT 1
9,8,'Warp Factor = ",wf  GO TO 8
10
 830 IF b$="w' OR b$="W" THEN GO
 TO 1530
 840 IF b$= d' OR b$='D" THEN GO
 TO 1400
 850 IF FN h()-tim0)=20-rank*2 T
HEN GO TO 1130
 860 IF b$=CHR$ 13 THEN GO TO 55
0
 870 IF ist=0 THEN GO TO 810
 880 GO SUB 180
 890 REM
 900 REM *** Your move ***
 910 REM
 920 IF stk>=4 AND stk<=6 THEN L
ET d(2)=d(2)-(d(2)>0)*wf  LET qd
=1
 930 IF stk=6 OR stk=2 OR stk=10
 THEN LET d(1 =d(1 +(d(1 <17 *wf
  LET qd=2
 940 IF stk=1 OR stk=5 OR stk=9
THEN LET d(1)=d 1) -(d(1)>0)*wf
LET qd=3
 950 IF stk>=8 AND stk<=10 THEN
LET d(2)=d(2)+(d(2)<31 *wf  LET
qd=4
 960 IF qd=4 THEN PRINT AT d(1),
d(2), INK ink0,'■"
 970 IF qd=3 THEN PRINT AT d(1)
d(2), INK ink0  ■"
 980 IF qd=2 THEN PRINT AT d(1),
d(2), INK ink0  ■"
 990 IF qd=1 THEN PRINT AT d(1),
d(2); INK ink0,'■'
1000 GO SUB 210  LET c(21,1)=d(1
). LET c(21,2 =d(2  LET move=0
 LET fuel=fuel-wf*wf  GO SUB 270
1010 LET l(31)=FN c(d(1),d 2),+1
1020 REM
1030 REM *** Ck Collision ***
1040 REM
1050 FOR i=20 TO 30  GO SUB 210
 IF l(31)=l(i) THEN CLS   PRINT
"********************************
* You have just run into a Star
  and were Vaporized!
  The Federation is doomed.
 ********************************
*"  GO TO 3940
1060 NEXT i
1070 FOR j=1 TO je  LET i=g(j)
IF l(31)=l(i) THEN GO TO 1090
1080 NEXT j  GO TO 1130
1090 GO SUB 2620  LET je=je-1  F
OR k=i TO je  LET g(k)=g(k+1)  N
EXT k
1100 REM
1110 REM *** Move Aliens ***
1120 REM
1130 GO SUB 180  FOR j=1 TO je
LET i=g(j). IF i>=6 AND i<=10 TH
EN GO SUB 210  IF l(i)>0 THEN PR
INT AT b(i,1 ,b(i,2)  INK ink3 "
■  LET b i,1)=b(i,1)+(c(b1,1)-b
(i,1) *RND/2  LET b(i,2)=b i,2)+
(c(b1,2)-b(i,2  *RND/2  PRINT AT
 b(i,1),b(i,2 , INK ink5, ■  LE
T l(i)=FN c(INT (b(i,1)+0.5),INT
 (b(i,2)+0.5))+1
1140 IF i>=1 AND i<=5 THEN GO SU
B 210  IF l(i  0 THEN PRINT AT b
(i,1) b(i 2 , INK ink3 '■  LET
b(i,1)=b(i,1)+ (c(b2,1)-b i,1)*R
ND/2  LET b(i,2)=b(i,2)+(c(b2,2)
-b(i 2 )*RND/2  PRINT AT b(i,1),
b(i,2 , INK ink5  ■  LET l(i)=F
N c(INT (b(i,1)+0.5) INT (b(i,2
+0.5))+1)
1150 REM
1160 REM *** Ck Collision ***
1170 REM
1180 IF l i)=l(31) THEN GO SUB 2
620  PRINT AT d(1),d(2), INK ink
0,"■"  LET je=je-1  FOR k=i TO j
e  LET g(k)=g(k+1)  NEXT k
1190 REM
1200 REM *** Star Base Stats ***
1210 REM
1220 IF i>=6 AND i<=10 THEN GO S
UB 210  IF l(i)=l(b1+10) THEN LE
T l(b1+10)=0  PRINT AT 20,0,'Sta
r Base ' b1,' has been destroyed
.'  PAUSE 60  LET b1=b1+1  LET s
(1)=0  LET s 2 =0
1230 IF i>=1 AND i<=5 THEN GO SU
B 210  IF l(i) = l(b2+10)  THEN LET
 l(b2+10) 0  PRINT AT 20 0, Star
 Base  ,b2," has been destroyed,
"  PAUSE 60  LET b2=b2-1  LET s(
1)=0  LET s(2)=0
1240 NEXT j
1250 IF l(19)=0 THEN LET b1=21
LET b2=21
1260 IF b2<b1 AND b1 <10 THEN LE
T b2=9  LET b1=9
1270 IF b1=10 THEN LET b2=21  LE
T b1=21
1280 IF b2=8 AND b1>=9 THEN LET
b1=21  LET b2=21
1290 REM
1300 REM *** Alien Attack ***
1310 REM
1320 LET x2=FN d(l(31))  LET y2=
FN e(l(31)): LET hit=0  LET dmg=
0  FOR j=1 TO je  LET i=g    GO
 SUB 210  IF ABS (b i,1)-d(1) )<=
(1+rank/6) AND ABS  b i 2)-d 2 )
 <=(1+rank/6) THEN LET hit=hit+1
 LET dmg=dmg+RND*.5*hit  LET x1=
FN d(l(i))  LET y1 FN e(l i    P
LOT INK ink5,x1 y1  DRAW INK ink
5,(x2-x1 ,(y2-y1)  BEEP .3 45  P
LOT INK ink3 x1,y1  DRAW INK ink
3,(x2-x1 ,(y2-y1   PRINT AT b(i
1),b(i,2), INK ink5, ■  PRINT A
T d 1 ,d 2), INK ink2  ■
1330 NEXT j  IF hit 0  THEN GO SU
B 180  PRINT AT 20 0 'Damage Con
trol  ' hit  hits on Shield
1340 LET sh=sh-dmg  IF sh <0  THE
N CLS  PRINT "*****************
**************** Your Shields hav
e gone down.    Emergency Proced
ures are in     effect.
                        Abandon S
hip '!'         *****************
****************        ou succes
sfully destroyed ',score  , enem
y ships. Thankyou for a goodgame
!",  ,"************************
*********Total mission time  ",FN
 h() ," seconds",' "*************
*********************'  PAUSE 0  G
O TO 3510
1350 IF hit>0 AND sh<5 THEN LET
sn=sh*.6
1360 IF hit>0 AND sh<5 THEN LET
mwf=INT (sh*.4+1
1370 GO SUB 350: IF hit>0 THEN P
AUSE 200
1380 IF move THEN GO TO 650
1390 GO TO 750
1400 REM
1410 REM *** Star Base Dock ***
1420 REM
1430 GO SUB 180  LET move=1: GO
SUB 210: FOR i=11 TO 19  IF l(31
)=l(i) THEN GO TO 1450
1440 NEXT i  GO SUB 200  PRINT A
T 20,0,"You are not in a Star Ba
se orbit", FLASH 1,AT 21,4 'Dock
ing is not possible!"  GO SUB 21
0  PAUSE 90  GO SUB 210  PAUSE 9
0: GO SUB 210: GO SUB 180  GO SU
B 350  GO TO 1130
1450 GO SUB 210  LET s(1)=d(1)
LET s(2)=d(2)  FOR j=1 TO je  LE
T i=g(j)  GO SUB 210  IF (ABS (b
(i,1 -d(1))+ABS (b(i,2 -d(2))) =
(2+rank-1) THEN GO SUB 200  PRIN
T AT 20,0,"Star Base will not lo
wer their  shields because there
 are enemy"  PRINT #0,AT 0,0   s
hips in the quadrant !'  GO SUB
210: PAUSE 90: GO SUB 210  PAUSE
 90: GO SUB 200  GO SUB 350  GO
TO 1130
1460 NEXT j  IF dock=1 THEN GO S
UB 200. PRINT AT 20,3;'Docking o
rbit is verified.",TAB 6,"Supply
 is initiated,"  LET fuel=fuel+I
NT (RND*40)  LET sh=sh+INT (RND*
20)  LET sn=sn+INT (RND*8)  LET
pt=pt+INT (RND*10)  LET ph=ph+IN
T (RND*10): GO SUB 210  GO SUB 1
510  GO SUB 210  PAUSE 60: GO SU
B 200  GO SUB 270  LET s(1)=d(1)
  LET s(2)=d(2)  GO TO 1130
1470 IF dock<1 THEN LET dock=doc
k+1: GO SUB 200: PRINT AT 20 0,
Docking orbit is verified.
Star Base refuses to lower it's'
  PRINT #0,AT 0,0,"Shields, Supp
ly is impossible. ": GO SUB 210
 PAUSE 60  GO SUB 210  PAUSE 60
 GO SUB 210  GO SUB 180  GO SUB
350  GO TO 1130
1480 REM
1490 REM *** Ck max load ***
1500 REM
1510 IF fuel>20 THEN LET fuel=20
1520 LET mwf=3
1530 IF sh>msh THEN LET sh=msh
1540 IF sn>msn THEN LET sn=msn
1550 IF pt>mpt THEN LET pt=mpt
1560 IF ph>mph THEN LET ph=mph
1570 RETURN
1580 REM
1590 REM *** Weapons ***
1600 REM
1610 GO SUB 200  GO SUB 210  PRI
NT AT 20,8,"Weapon Control ",TAB
 1,"Prepare to select your targe
t."
1620 LET wpn=1  IF ph>pt THEN LE
T wpn=2
1630 REM
1640 REM *** Tactical Scrn ***
1650 REM
1660 LET loc=FN c(9,16)+1  LET x
1=FN d(loc)  LET y1=FN e(loc)  L
ET x2=x1  LET y2=y1
```

Continued Next Page.

STARFLEET 2068

```
1670 LET js=0: LET d$=c$: FOR i=20 TO 30: GO SUB 210: LET f(i,1)=INT ((FN a(l(i))-d(1))*3+9): LET f(i,2)=INT ((FN b(l(i))-1-d(2))*3+16): IF f(i,1)>=0 AND f(i,1)<=18 AND f(i,2)>=0 AND f(i,2)<=30 THEN LET locs=FN c(f(i,1),f(i,2))+1: IF locs>0 AND locs<512 THEN LET d$(locs)="■": IF js<10 THEN LET js=js+1: LET g(js+10)=i
1680 NEXT i
1690 PRINT AT 0,0; INK ink7;d$(1 TO 512);TAB 31;" ";TAB 31;" "
1700 PRINT AT 9,16; INK ink0;"■": LET rng=24*sr: CIRCLE INK 9,x1,y1,rng: GO SUB 190: PRINT AT 21,12; FLASH 1;"Working": GO SUB 220
1710 REM
1720 REM *** Count Aliens ***
1730 REM
1740 LET move=1: LET irng=0: FOR i=1 TO 10: GO SUB 220: LET f(i,1)=INT ((FN a(l(i))-d(1))*3+9): LET f(i,2)=INT ((FN b(l(i))-1-d(2))*3+16): IF f(i,1)>0 AND f(i,1)<18 AND f(i,2)>0 AND f(i,2)<32 THEN LET irng=irng+1: LET g(irng)=i
1750 NEXT i
1760 REM
1770 REM *** Bases & Stars ***
1780 REM
1790 LET jb=0: FOR i=11 TO 19: GO SUB 220: LET f(i,1)=INT ((FN a(l(i))-d(1))*3+9): LET f(i,2)=INT ((FN b(l(i))-1-d(2))*3+16)-1: IF f(i,1)>0 AND f(i,1)<18 AND f(i,2)>0 AND f(i,2)<32 THEN LET jb=jb+1: LET g(jb+20)=i
1800 NEXT i
1810 LET loc=FN c(9,16)+1
1820 LET x1=FN d(loc): LET y1=FN e(loc): LET x2=x1: LET y2=y1: PRINT AT 9,16; INK ink0;"■"
1830 GO SUB 220: GO SUB 140: PRINT AT 17,1;"Tactical ";irng;" ships on screen.": GO SUB 350: LET tim0=FN h(): GO SUB 2090
1840 CIRCLE INK ink0,x2,y2,1: LET rx=FN f(y2): LET cx=FN g(x2): PRINT AT rx,cx; PAPER ink3;" ": PRINT AT 9,16; INK ink0;"■": IF FN h()-tim0>20-rank*2 THEN GO SUB 2940: GO TO 2900
1850 FOR j=1 TO irng: GO SUB 220: PRINT AT f(g(j),1),f(g(j),2); INK ink5;"■": NEXT j: FOR j=1 TO js: GO SUB 220: PRINT AT f(g(j+10),1),f(g(j+10),2); INK ink7;"■": NEXT j: FOR j=1 TO jb: GO SUB 220: PRINT AT f(g(j+20),1),f(g(j+20),2); INK ink1;"■": NEXT j
1860 REM
1870 REM *** Select Target ***
1880 REM
1890 LET timer=FN h()+20-2*rank
1900 RANDOMIZE USR 61241: PRINT AT FN f(y2),FN g(x2); OVER 1;"■"
1910 GO SUB 220: IF FN h()>timer THEN LET b$="N": GO TO 2070
1920 RANDOMIZE USR 61253: LET b$=INKEY$: IF b$<>CHR$ 32 THEN GO SUB 2050: GO SUB 2090: GO TO 1920
1930 GO SUB 500: IF b$<>CHR$ 13 AND stk=0 THEN PRINT AT FN f(y2),FN g(x2); OVER 1;"■": PAUSE 10: GO TO 1910
1940 IF stk>=4 AND stk<=6 THEN LET x2=x2-8*(x2>4)
1950 IF stk=2 OR stk=6 OR stk=10 THEN LET y2=y2-8*(y2>57)
1960 IF stk=1 OR stk=5 OR stk=9 THEN LET y2=y2+8*(y2<171)
1970 IF stk>=8 AND stk<=10 THEN LET x2=x2+8*(x2<250)
1980 PRINT AT FN f(y2),FN g(x2); OVER 1;"■"
1990 IF b$<>CHR$ 13 THEN GO TO 1920
2000 RANDOMIZE USR 61253: LET rx=FN f(y2): LET cx=FN g(x2): IF x1=x2 AND y1=y2 THEN GO SUB 180: GO SUB 2940: GO TO 2900
2010 GO SUB 200: GO SUB 550: GO TO 2150
2020 REM
2030 REM *** Select Weapon ***
2040 REM
2050 IF b$="T" OR b$="t" THEN LET wpn=1
2060 IF b$="P" OR b$="p" THEN LET wpn=2
2070 IF b$="N" OR b$="n" THEN GO SUB 190: PRINT AT 19,0;"Engineering Prepare Warp Drive": GO SUB 390: GO TO 750
2080 RETURN
2090 IF wpn=1 THEN PRINT AT 18,5; INVERSE 1;"Arming Photon Torpedo "
2100 IF wpn=2 THEN PRINT AT 18,5; INVERSE 1;"Charging Phaser Weapon"; INVERSE 0;TAB 31;" "
2110 RETURN
2120 REM
2130 REM *** Fire Weapon ***
2140 REM
2150 LET fuel=fuel-.2: IF fuel<=0 THEN GO TO 270
2160 LET range=SQR ((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2))/24-RND*.5
2170 GO SUB 220: IF range>sn THEN GO SUB 190: PRINT AT 19,0;"Weapons can not lock onto target! Target is out of range.": GO SUB 220: PAUSE 90: GO SUB 220: PAUSE 60: GO SUB 2940: GO TO 2900
2180 IF wpn=2 THEN GO TO 2260
2190 REM
2200 REM *** Photon Torpedo ***
2210 REM
2220 GO SUB 220: LET rx0=2*RND: IF pt<=0 THEN GO SUB 190: PRINT AT 18,4;"Torpedo bank is expended!": GO SUB 220: PAUSE 60: GO SUB 220: GO SUB 2940: GO TO 2900
2230 LET pt=pt-1: LET npts=INT (range*5): IF npts<=0 THEN LET npts=1
2240 GO SUB 190: LET ptr=1/npts: LET dix=(x2-x1)/npts: LET diy=(y2-y1)/npts: FOR i=1 TO npts: GO SUB 220: LET px=x1+INT (dix*i): LET py=y1+INT (diy*i): PLOT INK 9,px,py: BEEP .02,50-i*2: INVERSE 1: PLOT px,py: INVERSE 0: PRINT AT 9,16; INK ink0;"■": NEXT i
2250 GO TO 2370
2260 REM
2270 REM *** Phaser Weapon ***
2280 REM
2290 GO SUB 220: LET rx0=1: IF ph<=0 THEN GO SUB 190: PRINT AT 18,5;"Phaser bank is expended!": GO SUB 220: PAUSE 60: GO SUB 220: GO SUB 2940: GO TO 2900
2300 LET ph=ph-1
2310 GO SUB 190: PLOT INK ink0,x1,y1: DRAW INK 9,(x2-x1),(y2-y1): BEEP 1,30
2320 GO SUB 220: INVERSE 1: PLOT x1,y1: DRAW (x2-x1),(y2-y1): INVERSE 0
2330 PRINT AT 9,16; INK ink0;"■"
2340 REM
2350 REM *** Hit Something? ***
2360 REM
2370 LET rx=FN f(y2): LET cx=FN g(x2): LET dmg=0: LET hit=0: FOR j=1 TO (irng+js+jb): GO SUB 220: LET i=g(j): IF j>irng THEN LET i=g(10+j-irng)
2380 IF j>(irng+js) THEN LET i=g(20+j-irng-js)
2390 IF ABS (rx-f(i,1))<1 AND ABS (cx-f(i,2))<1 THEN LET hit=1: GO TO 2410
2400 NEXT j
2410 GO SUB 190: IF hit=0 THEN PRINT AT 18,8;"Target was missed!": GO SUB 220: PAUSE 60: GO SUB 220: PAUSE 60: GO SUB 2940: GO TO 2900
2420 IF i<=10 THEN PRINT AT 18,8;"Target was hit!": LET dmg=15+200*RND-(range-rx0*RND)*30-score/2: IF dmg<0 THEN LET dmg=0
2430 IF dmg>=100 THEN LET dmg=100
2440 LET f(i,3)=f(i,3)+dmg: IF f(i,3)<100 THEN GO TO 2510
2450 PRINT AT f(i,1),f(i,2); PAPER ink3;" ": LET f(i,3)=100: GO SUB 230: IF score=9 OR score=19 OR score=29 OR score=39 OR score=49 OR score=59 THEN GO TO 2710
2460 LET score=score+1: LET b(i,1)=0: LET b(i,2)=0: LET l(i)=0: IF i<=10 THEN LET j1=1: FOR j=1 TO irng: IF g(j)<>i THEN LET g(j1)=g(j): LET j1=j1+1
2470 GO SUB 220: NEXT j: LET irng=irng-1
2480 REM
2490 REM *** Hit Star Base ***
2500 REM
2510 IF i>=11 AND i<=19 THEN PRINT AT 19,1;" You just shot a Star Base! ": LET dock=-rank: PAUSE 100: LET l(i)=0: LET b(i-10,1)=0: LET b(i-10,2)=0: PRINT AT f(i,1),f(i,2);" ": GO SUB 2940: LET b1=21: LET b2=21: GO TO 2900
2520 REM
2530 REM *** Hit a Star ***
2540 REM
2550 IF i>=20 AND i<=30 THEN PRINT AT f(i,1),f(i,2); INK ink7;"■": PRINT AT 19,0;"You just shot a Star. The solar flare destroyed your sensors!   ": GO SUB 220: CIRCLE INVERSE 1,x1,y1,24*sn: PAUSE 60: GO SUB 220: PAUSE 60: LET sn=0: GO SUB 2940: GO TO 2900
2560 REM
2570 REM *** Hit Alien Ship ***
2580 REM
2590 PRINT AT 19,6;"Target Damage: ";INT (f(i,3)*10)/10;" %": IF f(i,3)<100 THEN PRINT AT rx,cx; INK ink5;"■": PAUSE 120: GO SUB 2940: GO TO 2900
2600 PRINT FLASH 1;AT 20,8;"Target was destroyed.": LET l(i)=0: LET f(i,1)=0: LET f(i,2)=0: PAUSE 200: GO SUB 200: GO SUB 2940
2610 GO TO 2900
2620 REM
2630 REM *** Collision Dmg ***
2640 REM
2650 GO SUB 180: PRINT AT 19,9; FLASH 1;"Collision"; FLASH 0;TAB 31;" You were hit by an enemy ship!": PAUSE 120: GO SUB 190: LET sh=sh-10*RND: IF sh<5 AND sn<0 THEN LET sn=.6*sh
2660 IF sh<0 THEN LET sh=0
2670 IF sn<0 THEN LET sn=0
2680 GO SUB 350: IF sh<=0 THEN GO TO 3920
2690 IF score=9 OR score=19 OR score=29 OR score=39 OR score=49 OR score=59 THEN GO TO 2710
2700 LET score=score+1: LET b(i,1)=0: LET b(i,2)=0: LET l(i)=0: RETURN
2710 REM
2720 REM *** Got 10 more ***
2730 REM
2740 FOR i=1 TO 10: LET f(i,3)=0: NEXT i: LET rank=rank+1: CLS : IF score>=59 THEN GO TO 2890
2750 CLS : PRINT AT 2,0;"********************************** Congratulations!                    You have successfully defended the Federation against 10 enemy Star Ships. ";'';" For your great victory over the enemy, the Federation Council has promoted you to the rank of";'';TAB 10;r$((rank-1)*15+1 TO (rank*15));'';" Keep up the good work! The next wave of ships has already been spotted on the long range sensors.                    *********************************"
2760 PRINT AT 21,10; PAPER 0; INK 7; FLASH 1;"Press Enter": PAUSE 0: CLS
2770 REM
2780 REM *** Next Wave ***
2790 REM
2800 PRINT AT 5,0;"******************************** The Federation Council would like to inform you that your ship will receive new shields at your next Star Base stop. The last information received shows that the next wave of invading ships have improved their weapon range.                 Be Careful!           **************************************": PRINT AT 21,10; PAPER 0; INK 7; FLASH 1;"Press Enter": PAUSE 0: CLS
2810 RESTORE 3630: IF b1>10 OR b2>10 THEN LET b1=5: LET b2=6
2820 FOR i=1 TO 19: READ a: LET l(i)=a: NEXT i: LET b1=b1-(rank-1): LET b2=b2+(rank-1): IF b1<1 THEN LET b1=1
2830 IF b2>8 THEN LET b2=8
2840 FOR i=1 TO (b1-1): LET l(i+10)=0: NEXT i: FOR i=(b2+1) TO 8: LET l(i+10)=0: NEXT i: LET msh=10+(rank-1)*2: LET mpt=5+rank-1: LET mph=5+rank-1
2850 LET score=score+1: GO TO 640
2860 REM
2870 REM *** All Done! ***
2880 REM
2890 CLS : PRINT AT 2,0;"********************************** Congratulations!                    You have successflly defended the Federation against all of the invaders!                  Way to go hot shot!       *********************************Total mission time ";FN h();" seconds";'';"*********************************": PAUSE 0: GO TO 3510
2900 GO SUB 190: PRINT AT 16,0;TAB 31;" ": GO TO 1790
2910 REM
2920 REM *** Alien Weapons ***
2930 REM
2940 GO SUB 190: LET hit=0: LET dmg=0: FOR j=1 TO irng: GO SUB 220: LET i=g(j): IF ABS (f(i,1)-9)<=(3.85+rank) AND ABS (f(i,2)-16)<=(3.85+rank) THEN LET hit=hit+1: LET dmg=dmg+RND+rank/6+(hit-1)*rank/30: LET locx=FN c(f(i,1),f(i,2))+1: LET x3=FN d(locx): LET y3=FN e(locx): PLOT INK ink5,x3,y3: DRAW INK ink5,(x1-x3),(y1-y3): BEEP .3,45: INVERSE 1: PLOT x3,y3: DRAW (x1-x3),(y1-y3): INVERSE 0: PRINT AT f(i,1),f(i,2); INK ink5;"■": PRINT AT 9,16; INK ink0;SCREEN$ (9,16)
2950 NEXT j: IF hit>0 THEN GO SUB 180: PRINT AT 19,0;"Damage Control ";hit;" hits on shield": GO SUB 350
2960 LET sh=sh-dmg: IF sh<=0 THEN CLS : PRINT AT 10,0;"********************************** Your shields have gone down.   Emergency procedures are in effect.                        Abandon Ship !!!          ****************************************": GO TO
```

3940

```
2970 REM
2980 REM *** Alien Tactical ***
2990 REM
3000 IF hit>0 AND sh<5 AND sn<>0
 THEN CIRCLE INVERSE 1,x1,y1,24*
sn. IF sn<>0 THEN LET sn sh*.6
GO SUB 360  CIRCLE x1 y1,24*sn
3010 LET ir1=1
3020 REM
3030 REM *** Ships 1 & 6 ***
3040 REM
3050 FOR j=ir1 TO irng  GO SUB 2
20  LET i=g(j)  IF l(i) 0 AND (i
=1 OR i=6) AND (f(i,1)>=0 AND f(
i,1)<=18) AND (f(i,2 =0 AND f(i
,2)<=30) THEN PRINT AT f(i,1),f(
i,2), PAPER ink3," "  LET f(i,1
=INT (f(i,1)+(7 SGN (RND-.5)*1-r
ank/2-f(i,1))/2)  LET f(i,2) INT
 (f(i,2)+(14-SGN (RND-.5 *1-rank
/2 f(i,2))/2)  PRINT AT f(i,1),f
(i,2), INK ink5,"█"  LET xr=INT
((f(i,1)-9 /3)+d 1)  LET xc=INT
((f(i,2)-16)/3)+d(2)  LET l(i)=F
N c(xr xc)+1
3060 REM
3070 REM *** Ships 2 & 7 ***
3080 REM
3090 IF l(i)>0 AND (i=2 OR i=7)
AND (f(i,1)>=0 AND f i,1 <=18) A
ND (f(i,2)>=0 AND f(i,2 <=30  TH
EN PRINT AT f(i,1),f i 2), PAPER
 ink3, "  LET f(i,1)=INT (f(i,1
)+(7-SGN (RND-.5)*1-rank/2-f(i,1
))/2)  LET f(i,2)=INT (f(i,2)+(1
8+SGN (RND-.5)*1+rank/2-f(i,2))/
2)  PRINT AT f(i,1 ,f(i 2), INK
ink5,"█"  LET xr=INT ((f(i,1)-9
/3)+d(1)  LET xc=INT ((f(i,2)-16
)/3)+d(2). LET l(i)=FN c(xr,xc)+
1
3100 REM
3110 REM *** Ships 3 & 8 ***
3120 REM
3130 IF l(i)>0 AND (i=3 OR i=8)
AND (f(i,1)>=0 AND f i,1 <=18) A
ND (f(i,2)>=0 AND f(i,2)<=30) TH
EN PRINT AT f(i,1),f i,2 , PAPER
 ink3;" "  LET f(i,1)=INT (f(i,1
)+(13+SGN (RND-.5)*1+rank/2-f(i,
1))/2)  LET f(i,2)=INT (f(i,2)+(
14+SGN (RND-.5)*1+rank/2-f(i,2)
/2)  PRINT AT f(i,1),f(i,2), INK
 ink5,"█"  LET xr=INT ((f(i,1) 9
)/3)+d(1)  LET xc=INT ((f(i,2)-1
6)/3)+d(2   LET l(i)=FN c(xr,xc)
+1
3140 REM
3150 REM *** Ships 4 & 9 ***
3160 REM
3170 IF l(i)>0 AND (i=4 OR i=9)
AND (f(i,1)>=0 AND f(i,1)<=18) A
ND (f(i,2)>=0 AND f(i,2)<=30) TH
EN PRINT AT f(i,1),f(i,2); PAPER
 ink3;" "  LET f(i,1)=INT (f(i,1
)+(13+SGN (RND-.5)*1+rank/2-f(i,
1))/2): LET f(i,2)=INT (f(i,2)+(
18+SGN (RND-.5)*1+rank/2 f(i,2))
/2)  PRINT AT f(i,1),f(i,2), INK
 ink5,"█"  LET xr=INT ((f(i 1)-9
)/3)+d(1)  LET xc=INT ((f(i 2)-1
6)/3)+d(2): LET l(i)=FN c(xr,xc)
+1
3180 REM
3190 REM *** Ships 5 & 10 ***
3200 REM
3210 IF l(i)>0 AND (i=5 OR i=10)
 AND (f(i,1)>=0 AND f(i,1)<=18)
AND (f(i,2)>=0 AND f(i,2)<=30) T
HEN PRINT AT f(i 1),f(i 2), PAPE
R ink3," ". LET f(i,1) INT (f(i,
1)+(9-SGN (RND-.5)*(3+rank/2)*RN
D-f(i,1))/2)  LET f(i,2)=INT (f(
i,2)+(16+(3+rank/2)*RND*SGN (RND
-.5)-f(i,2))/2)  PRINT AT f(i,1)
,f(i,2), INK ink5,"█"  LET xr=IN
T ((f(i,1)-9)/3)+d(1)  LET xc=IN
T ((f(i,2)-16)/3)+d(2)  LET l(i)
=FN c(xr,xc)+1
3220 IF f(i,1)<>9 OR f(i,2) <>16
THEN GO TO 3280
3230 GO SUB 220  GO SUB 2650  LE
T f(i,1)=0  LET f(i,2)=0  LET f(
i,3) 100  PRINT AT 9,16, INK ink
0,"█"  LET j1=1  FOR k=1 TO irng
  GO SUB 210  IF g(k)<>i THEN LE
T g(j1)=g(k). LET j1=j1+1
3240 NEXT k: LET irng=irng-1  LE
T ir1=j+1. GO SUB 180  GO TO 305
0
3250 REM
3260 REM *** Ck Collision ***
3270 REM
3280 FOR k=1 TO js  GO SUB 220
LET k10=k+10  IF f(i,1)=f(g(k10)
,1) AND f(i,2)=f(g(k10),2) THEN
GO TO 3350
3290 NEXT k
3300 REM
3310 REM *** Star Base Stats ***
3320 REM
3330 FOR k=1 TO jb  GO SUB 220
LET k10=k+20. IF f(i,1)=f(g(k10)
,1) AND f(i,2)=f(g(k10),2) THEN
GO TO 3370
3340 NEXT k: GO TO 3430
3350 GO SUB 180  PRINT AT 19,0,"
An enemy ship just flew into a
star and was vaporized!"  PRINT
AT f(i,1),f(i,2), INK ink7,"█"
GO SUB 230: PAUSE 120: IF score=
9 OR score=19 OR score=29 OR sco
re=39 OR score=49 OR score=59 TH
EN PAUSE 60  GO TO 2710
3360 GO TO 3440
3370 GO SUB 180  IF g(k10)>=11 A
ND g(k10) =15 THEN GO SUB 210  L
ET l(b1+10)=0  PRINT AT 20,0,"St
ar Base " b1," has been destroye
d.  LET s(1)=0  LET s(2)=0  PAU
SE 60  LET b1=b1+1
3380 IF g(k10)>=16 AND g(k10)<=2
0 THEN GO SUB 220  LET l b2+10 =
0  PRINT AT 20,0,"Star Base  ,b2
  has been destroyed."  LET s(1
=0. LET s(2)=0  PAUSE 60  LET b
2=b2-1
3390 LET jb=jb-1  IF l(19)=0 THE
N LET b1=21  LET b2=21
3400 IF b2<b1 AND b1<>10 THEN LE
T b2=9  LET b1=9
3410 IF b1=10 THEN LET b2=21  LE
T b1=21
3420 IF b2=8 AND b1>=9 THEN LET
b1=21  LET b2=21
3430 NEXT j  GO TO 3460
3440 LET score=score+1  LET b(i,
1)=0  LET b(i,2 =0  LET f(i,1)=0
  LET f(i,2)=0  LET l(i)=0  LET
j1=1  FOR k=1 TO irng  IF g(k)<>
i THEN LET g(j1)=g(k). LET j1=j1
+1
3450 NEXT k: LET irng=irng-1  LE
T ir1=j+1  GO SUB 180  GO TO 305
0
3460 RETURN
3470 STOP
3480 REM
3490 REM *** S T A R T ***
3500 REM
3510 CLEAR 61240. DIM t(15)  DIM
 a$(704)  DIM b$ 1)  DIM b(10,2)
  DIM c(21,2)  DIM c$(512)  DIM
d(2)  DIM d$(512)  DIM e(10,2)
DIM f$(20)  DIM f(30,3)  DIM g(3
0)  DIM l(31)  DIM p(10)  DIM r$
(90): DIM s(2): DIM s$(132)  RES
TORE
3520 REM
3530 REM *** M/C Routines ***
3540 REM
3550 DATA INT 17,81,239,33,0,64,
1,0,16,237,176,201,17,0 64,33,81
,239,1,0,16,237,176,201  FOR i=6
1241 TO 61264: READ j. POKE i,j.
 NEXT i
3560 REM
3570 REM *** UDG Data ***
3580 REM
3590 DATA INT 0,12,0,48,128,66,1
6,255,1,0,128,0,24,24,0,24,70,12
6,16,0,1,0,128,0 60,48,129,12,42
,126 24,0,1,0,128,66,126,120,219
,30,20,90,63,0,1 0,128,129,219,1
20,126,30,40,90,252,0,1,0,128,12
9,129 48,60,12,84,24,24,0,1,0,12
8,66 0,24,24,24,98,24,8,0,1,0,12
8,0,0,12,0,48,1,0,8,0,1,255,128,
0
3600 REM
3610 REM *** Init. Position ***
3620 REM
3630 DATA INT 7,7,7,7,7,25,25,25
,25,25,80,150,282,406,464,394,26
2,138,272
3640 RANDOMIZE
3650 REM
3660 REM *** Init. Variables ***
3670 REM
3680 LET qd=1  LET score=0  LET
ink0=7  LET ink1=6. LET ink2=2
LET ink3=1  LET ink4=4  LET ink5
=5  LET ink6=6  LET ink7=7  LET
r$(1 TO 90)="Captain         Comm
odore      Rear Admiral   Vice A
dmiral   Admiral          Fleet Ad
miral  "  BORDER 1  PAPER 1  INK
 7
3690 CLS   LET t(1)=0: LET t(2)=
0  LET t(3)=0  LET t(4)=0. LET t
(5)=0  LET t(6)=0  LET t(7)=15
LET t(8)=7  LET t(9)=16  LET t(1
0)=16  LET t(11)=16  LET t(12)=1
0  LET t(13)=16. LET t(14)=13

3700 REM
3710 REM *** Poke UDG ***
3720 REM
3730 FOR i=0 TO 7
3740 READ g  POKE USR "a"+i,g
3750 READ g  POKE USR "b"+i,g
3760 READ g  POKE USR "c"+i g
3770 READ g  POKE USR "d"+i,g
3780 READ g  POKE USR "e"+i g
3790 READ g  POKE USR "f"+i,g
3800 READ g. POKE USR "g"+i,g
3810 READ g  POKE USR "h"+i,g
3820 READ g. POKE USR "i"+i,g
3830 READ g  POKE USR "j"+i,g
3840 READ g. POKE USR "k"+i,g
3850 READ g  POKE USR "m"+i,g
3860 NEXT i
3870 INK 9. PRINT AT 21,13  PAPE
R 0, INK 7, FLASH 1,"Working   L
ET rank=1  PRINT AT 5,0," Alert!
  Alert!  Alert!  Alert! ",,"",
 Star Cruisers have invaded the
Federation's territory. You are
Captain of the Star Ship Strata-
Gem. You must protect all of the
Federation Star Bases. You are
our last hope!",,
Good Luck!"
```

STARFLEET 2068

```
3880 FOR i=1 TO 19  READ a  LET
l(i)=a: NEXT i  FOR i=20 TO 30
LET cx=14+22* RND-.5)  LET rx=9+
13*(RND .5)  LET l(i)=FN c rx cx
)+1  NEXT i  LET l(31) 271  LET
fuel=20. LET msh=10  LET sh=msh
 LET msn=3. LET sn=msn  LET mpt=
5: LET pt=mpt: LET mph=5. LET ph
=mph. LET mwf=3
3890 LET wf=1. FOR i=11 TO 19  F
OR j=20 TO 30  IF l(j)=l(i) THEN
 LET l(j)=l(j)+1
3900 NEXT j  NEXT i
3910 FOR i=1 TO 32  LET a$(i =" █
"  NEXT i  FOR i=545 TO 578  LET
 a$(i)="█"  NEXT i  FOR i=0 TO 1
7  LET a$(FN c(i,0)+1)="█"  LET
a$(FN c(i,31)+1)  "█"  NEXT i  LE
T c$(1 TO 480)=a$(1 TO 480  FOR
 i=481 TO 512  LET c$(i)="█"  NE
XT i. FOR i=20 TO 30  LET a$ (l(i
))="█"  NEXT i
3920 FOR i=1 TO 30  LET f(i,1)=0
  LET f(i,2)=0  LET f(i 3)=0  NE
XT i. LET move=1  LET dock=1  LE
T s(1)=0  LET s(2)=0  LET b1=1
LET b2=8: BEEP ,25 30  POKE 2367
4,0. POKE 23673,0. POKE 23672,0
 GO TO 3960
3930 CLS
3940 PRINT AT 0,0,"***************
************************ You have suc
cessfully destroyed";INT score,"
 enemy ships!",;"Thankyou for a
 good game!        Press Enter to
 continue.        ****************
**********************Total mission
time: ";FN h();" seconds",,"***
**********************************"
PAUSE 0  IF INKEY$=CHR$ 13 THEN
CLS   GO TO 3510
3950 REM *** get ready ***
3960 CLS   PRINT AT 10,12, PAPER
 2; INK 7, FLASH 1,"Alert!"
3970 SOUND 7,62,8,15
3980 FOR j=1 TO 3  FOR i=250 TO
120 STEP -5: SOUND 0,i  NEXT i
PAUSE 4. NEXT j
3990 PRINT AT 13,7, PAPER 2, INK
 7, FLASH 1,"Battle Stations!"
4000 FOR j=1 TO 3: FOR i=250 TO
120 STEP -5  SOUND 0,i  NEXT i
PAUSE 4: NEXT j
4010 SOUND 8,0;9,0,10,0
4020 POKE 23674,0  POKE 23673,0
 POKE 23672,0
4030 GO TO 640
4040 SAVE "Star Fleet"
```

CK Type

Line	Bytes	Sum
10	25	1912
20	40	3350
30	25	1912
40	93	7873
50	25	1912
60	26	2050
70	29	2259
80	36	2288
90	34	2149
100	36	2830
110	37	2441
120	25	1926
130	80	5720
140	23	2343
150	2	481
160	24	2969
170	2	481
180	63	3677
190	91	5175
200	118	6635
210	36	3162
220	51	3878
230	338	12888
240	2	481
250	21	2552
260	2	481
270	213	21051
280	130	10872
290	77	5687
300	87	6126
310	76	6164
320	2	481
330	24	3066
340	2	481
350	80	6349
360	127	8645
370	2	481
380	2	481
390	310	21339
400	27	2276
410	134	12087
420	3	617
430	115	8858
440	3	617
450	34	2880
460	11	750
470	2	481
480	22	2761
490	2	481
500	81	7728
510	21	2490
520	2	481
530	21	2489
540	2	481

STARFLEET 2068

550	73	7871
560	97	7176
570	84	9730
580	16	2149
590	48	5505
600	49	6344
610	2	481
620	16	2072
630	2	481
640	36	3806
650	11	917
660	108	8156
670	11	934
680	6	1058
690	24	3156
700	27	2842
710	27	2842
720	26	3158
730	35	3362
740	29	2574
750	2	481
760	20	2509
770	2	481
780	16	3049
790	11	917
800	134	11807
810	208	16509
820	84	8366
830	27	2842
840	27	2842
850	45	4214
860	25	2389
870	24	2162
880	11	917
890	2	481
900	19	2312
910	2	481
920	94	6412
930	98	7195
940	55	6912
950	86	6731
960	46	4066
970	46	4059
980	46	4057
990	45	4054
1000	117	8444
1010	47	2520
1020	2	481
1030	22	2703
1040	2	481
1050	228	21468
1060	3	617
1070	52	5275
1080	15	1409
1090	57	5858
1100	2	481
1110	21	2493
1120	2	481
1130	358	26793
1140	324	23896
1150	2	481
1160	22	2703
1170	2	481
1180	100	10450
1190	2	481
1200	25	3013
1210	2	481
1220	201	16397
1230	200	16441
1240	3	618
1250	47	3383
1260	43	3933
1270	39	3191
1280	49	4079
1290	2	481
1300	22	2637
1310	2	481
1320	432	42002
1330	88	9308
1340	436	43645
1350	40	4767
1360	52	5453
1370	35	3021
1380	17	2230
1390	11	977
1400	2	481
1410	24	2854
1420	2	481
1430	67	7152
1440	209	19133
1450	396	34250
1460	340	31110
1470	277	26701
1480	2	481
1490	21	2479
1500	2	481
1610	30	3024
1520	13	1161
1530	16	3049
1540	16	3097
1550	16	3133
1560	16	3037
1570	2	521
1580	2	481
1590	17	2014
1600	2	481
1610	101	10279
1620	33	663
1630	2	481
1640	23	2818
1650	2	481
1660	72	7871
1670	390	29237
1680	3	617
1690	86	5721
1700	128	11031
1710	2	481
1720	22	2689
1730	2	481
1740	295	20720
1750	3	617
1760	2	481
1770	23	2571
1780	2	481
1790	291	18298
1800	3	617
1810	34	2077
1820	67	8259
1830	114	11283
1840	147	15550
1850	254	22141
1860	2	481
1870	23	2830
1880	2	481
1890	34	2991
1900	43	4943
1910	43	4697
1920	70	7206
1930	88	8616
1940	53	4837
1950	67	5702
1960	67	5818
1970	56	5418
1980	29	3421
1990	25	2724
2000	84	10334
2010	34	2862
2020	2	481
2030	23	2829
2040	2	481
2050	28	3207
2060	28	3199
2070	102	10315
2080	2	521
2090	66	6468
2100	89	7821
2110	2	521
2120	2	481
2130	21	2499
2140	2	481
2150	45	5900
2160	62	5603
2170	198	13720
2180	25	2385
2190	2	481
2200	24	3092
2210	2	481
2220	152	14120
2230	66	6896
2240	227	23978
2250	12	943
2260	2	481
2270	23	2847
2280	2	481
2290	149	13308
2300	15	1385
2310	68	6233
2320	53	4923
2330	29	2465
2340	2	481
2350	24	2911
2360	2	481
2370	122	14200
2380	38	4638
2390	80	6321
2400	3	618
2410	130	11539
2420	138	12248
2430	30	3334
2440	67	4837
2450	175	15444
2460	154	13678
2470	33	3582
2480	2	481
2490	23	2652
2500	2	481
2510	255	14086
2520	2	481
2530	20	2218
2540	2	481
2550	253	23603
2560	2	481
2570	24	2799
2580	2	481
2590	153	12143
2600	145	11278
2610	12	986
2620	2	481
2630	23	2796
2640	2	481
2650	184	17736
2660	24	2135
2670	24	2209
2680	35	3102
2690	102	10066
2700	81	6136
2710	2	481
2720	21	2266
2730	2	481
2740	94	8927
2750	499	52347
2760	72	5635
2770	2	481
2780	19	2230
2790	2	481
2800	445	48040
2810	61	4659
2820	102	9803
2830	24	2201
2840	189	13514
2850	32	3167
2860	2	481
2870	19	2046
2880	2	481

How does it work?

Here is a brief description of the major sections (and a little philosophy) behind what makes this program work. This won't tell you everything you'll need to know about writing a game program, nor everything there is to know about this program, but it will tell you enough to make changes to this one, and even cheat if you want too!

Lines 10 - 50: Title/Copyright notice. Be sure to include this

Lines 60 - 120: These calculations are used over and over in the program. To save memory, they are defined as functions. They are used to calculate row/column position from x/y pixel position and visa versa.

Line 130: This line reads the system 'frames' counter and calculates system time in seconds. It is used to calculate the GAME time, and also the elapsed time from your last move. If you take too much time to make a move, the aliens will take their turn. The time to make a turn is shorter as the level of difficulty goes up!

Line 140: This line re-directs the 'RUN' command to the program initialize routines starting at line 3510. (These are at the bottom/end of the program because they are not used very often, the routines that are used the most are at the top/beginning of the program to help speed things up')

Lines 150 - 220: These lines 'CLEAR' away old messages and print the Elapsed Game time.

Line 230: This sets the SOUND command for the 'soft' exploding ship.

Lines 240 - 310: This section checks how much fuel you have left. It also prints the 'fuel gage' marker in the appropriate color, GREEN for lots of fuel, YELLOW for mid-tank, RED for low!

Lines 320 - 360: These lines print the Weapons Status.

Lines 370 - 460: This section converts the ALIENS, STAR BASES, STARS, and YOUR SHIP coordinates from the Screen Location array (L) to their row/column coordinates. Then the Navigation screen is printed. NOTE: to check the position of a star, base, etc. against any other object, you must compare the row and column, or x and y positions, this is 2 coordinate comparisons per object. When you start checking 10 ships for collisions with 9 bases, 11 stars, and your ship, it becomes very time consuming. Therefore, I created a single array (L), that defines a combined Row/Column location as a single value, thus reducing the comparison time to half its former time! The time to calculate this is easily compensated because there are a lot more comparisons than calculation loops!

Lines 470 - 510: This section reads the joystick using an IN/OUT command. This simultaneously reads both joysticks and is much faster than a STICK command that reads left only or right only. This method allows you to use left or right joysticks without any program speed penalty!

Lines 520 - 610: 'LOCKING ONTO TARGET'' While this message is flashing, these lines are checking to 'see' if there are any objects between your ship and the selected target. This is done using the POINT (X,Y) command. If POINT is 1, there is an object at that X,Y location. If POINT is 0, no object. The first object encountered is set as the real 'target'.

Lines 620 - 740: This is the NAV. screen menu. Just press the letter associated with your choice: N - Nav, W - Weapons, D - Dock

Lines 750 - 880: 'HELM AT YOUR COMMAND'. It's your move!

Lines 890 - 1010: Move. Have enough fuel? Reposition your ship.

Lines 1020 - 1090: Check if you 'hit' anything when you moved.

Lines 1100 - 1140: Move the Aliens. 5 aliens move clockwise around the screen, 5 aliens move counter-clockwise.

Lines 1150 - 1280: Check if the Alien ships collided with your ship, or any of the Star Bases. The Star Bases are destroyed when the Alien Ships get close enough to 'dock'. When a Star Base is destroyed, the next Star Base becomes their target. When Star Base 9 (home base) is destroyed, your ship becomes the final target!

2890	341	32285
2900	56	4205
2910	2	481
2920	23	2776
2930	2	481
2940	479	44766
2950	98	8808
2960	251	22739
2970	2	481
2980	24	2942
2990	2	481
3000	121	11750
3010	13	1090
3020	2	481
3030	21	2026
3040	2	481
3050	549	37957
3060	2	481
3070	21	2023
3080	2	481
3090	518	34136
3100	2	481
3110	21	2230
3120	2	481
3130	519	33361
3140	2	481
3150	21	2032
3160	2	481
3170	519	33981
3180	2	481
3190	22	2118
3200	2	481
3210	527	34391
3220	56	4274

Lines 1280 - 1390: Alien attack. If you get too close to the aliens (even in Nav mode), they will attack you!

Lines 1400 - 1470: DOCK. Check if you are in Star Base orbit, if alien ships are too close, etc.. Begin refuel, repair, etc. if DOCKed.

Lines 1480 - 1570: This section limits the fuel, armament, etc. carried for each level of difficulty.

Lines 1580 - 1620: Prepare to select your target.

Lines 1630 - 1700: This section catalogs all the stars, aliens, and bases relative to your ships' present location.

Lines 1710 - 1850: This section counts how many stars, aliens, and bases are visible on the weapons screen, then creates the screen display.

Lines 1860 - 2010: This section reads the joystick commands and moves the 'cross-hair' UDG used to select the target. The M/C routine at 61241 makes a duplicate of the weapons screen in 'high' memory. The UDG cross-hair is moved to wherever you command it. The M/C routine at 61253 replaces the original screen, erasing the cross-hairs, allowing you to print it at another location quickly, giving you fast response to joystick commands.

Lines 2020 - 2110: This section allows you to select which weapon you want. It also prints your selection.

Lines 2120 - 2180: FIRE! Is your target in range? Do you have any fuel left?

Lines 2190 - 2250: Fire Photon Torpedo. Any left? Torpedo is displayed as a blinking 'dot' as is crosses the screen.

Lines 2260 - 2330: Fire Phaser. Any left? Phaser is displayed as a line.

Lines 2340 - 2610: Hit something? This section determines if you actually hit a anything, and what it was. If you hit an alien ship, its damage is computed. Alien ship damage is additive. You may score 30% on it with a first hit, 80% on a second hit.... the two hits will score a 'kill'. Did you hit a star? a star base?

Lines 2620 - 2700: This subroutine determines the outcome of a collision with an alien ship. The alien is destroyed, but what is the damage to your ship???

Lines 2710 - 2850: Got 10 more aliens! Level of difficulty is increased.

Lines 2860 - 2900: Victory! Got 'em all!

Lines 2910 - 2960: Aliens shoot at your ship. Damage?

Lines 2970 - 3240: When you are in the Weapons mode, the aliens try to surround you. This section of the program controls each of the alien ships, and places them at prescribed points around you with some random deviations from these locations. The exact positions also vary with the level of difficulty! Despite the complexity of each move, the wait is very short... less than a second per ship.

Lines 3250 - 3470: Check alien ship collisions: hit your ship? destroy a star base?

Lines 3520 - 4040: Program initialization section.

Line 3550: This is the data and read/poke routine to set up the machine code programs used with the weapon screen.

Line 3590: This is the data that defines all 13 UDG sprites.

Line 3630: This data defines the screen location (L) of the 9 star bases and 10 alien ships.

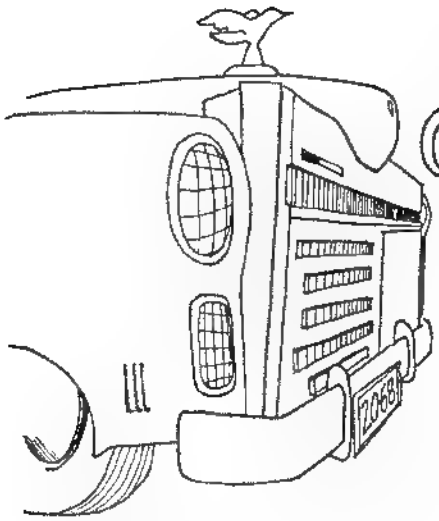
Lines 3700 - 3860: These lines POKE in the UDG values.

Line 3910: This line creates a$. This array contains the navigation screen definition: screen border, star location, etc. This string is printed, and gives the appearance of scrolling the map into place.

Line 4020: This line pokes the 'frames' system variable with 0 (zero), reinitializing the game timer.

I hope you enjoy playing Star Fleet... at least as much as I did writing it! OK, beam me up Scotty!

3230	194	16113	3370	202	15819	3510	276	18648
3240	60	6150	3380	191	15528	3520	2	481
3250	2	481	3390	62	4462	3530	22	2566
3260	22	2703	3400	43	3933	3540	2	481
3270	2	481	3410	39	3181	3550	258	14744
3280	116	8952	3420	49	4079	3560	2	481
3290	3	619	3430	15	1466	3570	18	1883
3300	2	481	3440	186	14855	3580	2	481
3310	25	3013	3450	60	6150	3590	354	30504
3320	2	481	3460	2	521	3600	2	481
3330	116	8972	3470	2	455	3610	24	2979
3340	15	1431	3480	2	481	3620	2	491
3350	260	26035	3490	19	1612	3630	176	7493
3360	12	1039	3500	2	481	3640	2	511
						3650	2	481
						3660	25	3020
						3670	2	481
						3680	281	25234
						3690	294	16684
						3700	2	481
						3710	18	1978
						3720	2	481
						3730	19	1297
						3740	13	1971
						3750	13	1973
						3760	13	1975
						3770	13	1977
						3780	13	1979
						3790	13	1981
						3800	13	1983
						3810	13	1985
						3820	13	1987
						3830	13	1989
						3840	13	1991
						3850	13	1993
						3860	3	617
						3870	342	37170
						3880	287	26378
						3890	84	7099
						3900	6	1174
						3910	277	23341
						3920	276	17970
						3930	2	515
						3940	318	32046
						3950	5	741
						3960	58	4356
						3970	35	1185
						3980	77	6148
						3990	55	4469
						4000	77	6143
						4010	50	1046
						4020	63	4068
						4030	11	804
						4040	14	1470



CLASSY FRONT END

Part II

BY PAUL BINGHAM

Judging by all the mail, it appears that a good many 2068 users have been interested in alternatives to Sinclair's original font for quite some time! And that's exactly what we will be discussing this time. Stay tuned!

Now, in all those letters no one mentioned a little bug I ran across in the BASIC program of Part I. That's too bad—I usually send such computer pest controllers a tape of utilities or at least a thank-you card!

The bug is a small one in line 61. This line is supposed to print an equal sign—I say supposed to. If one examines the new characters in Figure 3 in the same article he won't find an equal sign. Line 61 has "DRAW 0,2" twice. Change these to "DRAW 2,0" and the program will perform as intended. Soon we'll be looking at some slick code that will make worrying about such inherent hoohaa of BASIC a thing of the past.

So now that the bug is out of the way, most of my mail is answered and my new son's sleeping through the night, I will dive headlong into this next level of CLASSY FRONT END. We will try to progress slowly enough so as not to lose anyone. So even those less familiar with programming are invited to follow along.

In designing a Machine Code utility one faces the inevitable quandry of too many desires and not enough space. I wrote a wish list of features and then settled from that on a "short list" of must-haves. These fell into three catagories (being BASIC compatible, being flexible and easy to use). When we finally finish the program it will include the following features:

A. BASIC COMPATIBLE	1. reads coordinates/text from BASIC
	2. is callable from BASIC
	3. allows use of 2068 symbols, too
B. FLEXIBLE	4. able to reside anywhere in memory
	5. all symbols changible by the user
C. EASY TO USE	6. short enough to load quickly
	7 provides complete set of symbols
	8 runs much faster than BASIC
	9. keeps track of screen coordinates

One of the most constraining to achieve as it turns out is #4. Without fixed addresses for each byte the program has used two unused bytes at 23728 and 23729 (5CB0 and 5CB1 hex) to store it's location as a reference. But more on that later.

Some who wrote me asked about "printer compatability". As you will notice this does not appear on the "short list". Now CLASSY FRONT END is really a screen print environment, not another printer font. It is true that CLASSY is compatable with the 2040 printer—several letters I received were in NEW 2068 MEDIUM printed on thermal paper which thrilled me (thanks, guys!).

But opening the old can of printer-compatability-worms is not what I want to do. Lest some would feel hung out to dry, take heart that CLASSY keeps track of text in its own font as well as the Sinclair font. Because of this, any printer should be able to print in its own font style by using standard ASCII codes. And I'm sure you can, too. Just consult your interface and printer manuals.

Foremost in the design of the machine code program is how we will store as well as display the graphic data for our new font. Our 2068 we know has a character table in ROM starting at 15616 (3D00 hex). Here the 2068's 95 symbols are stored. The PRINT routine starting at 8537 (2159 hex) looks up the codes stored in this table for each symbol it prints.

It works something like this: first the routine finds exactly where the code for the actual symbol begins in the table. Next it reads the first of eight bytes of code. This is always a number between 0 and 255. Remember reading someplace that all bytes contain eight bits? It is true—by using this code the 2068 figures out which bits are set or not set and then the PRINT routine darkens in the corresponding eight pixels on the screen. By doing this with each of the succeeding of the eight bytes, the PRINT routine assembles a pixel pattern a line at a time on the screen. By this method the 2068 can store the graphic information for the 64 pixels of each symbol in only eight bytes! Clever these British...

To know just which code number represents which pixel pattern is fairly easy to figure out. The program in Listing #1 entitled "Z80 bit patterns" will let our 2068 do it for us. Be sure to have enough printer paper on hand as the 2040 printout is almost three feet long. (You will certainly need this if you do any alterations or customizing of the font as listed.)

Listing 1

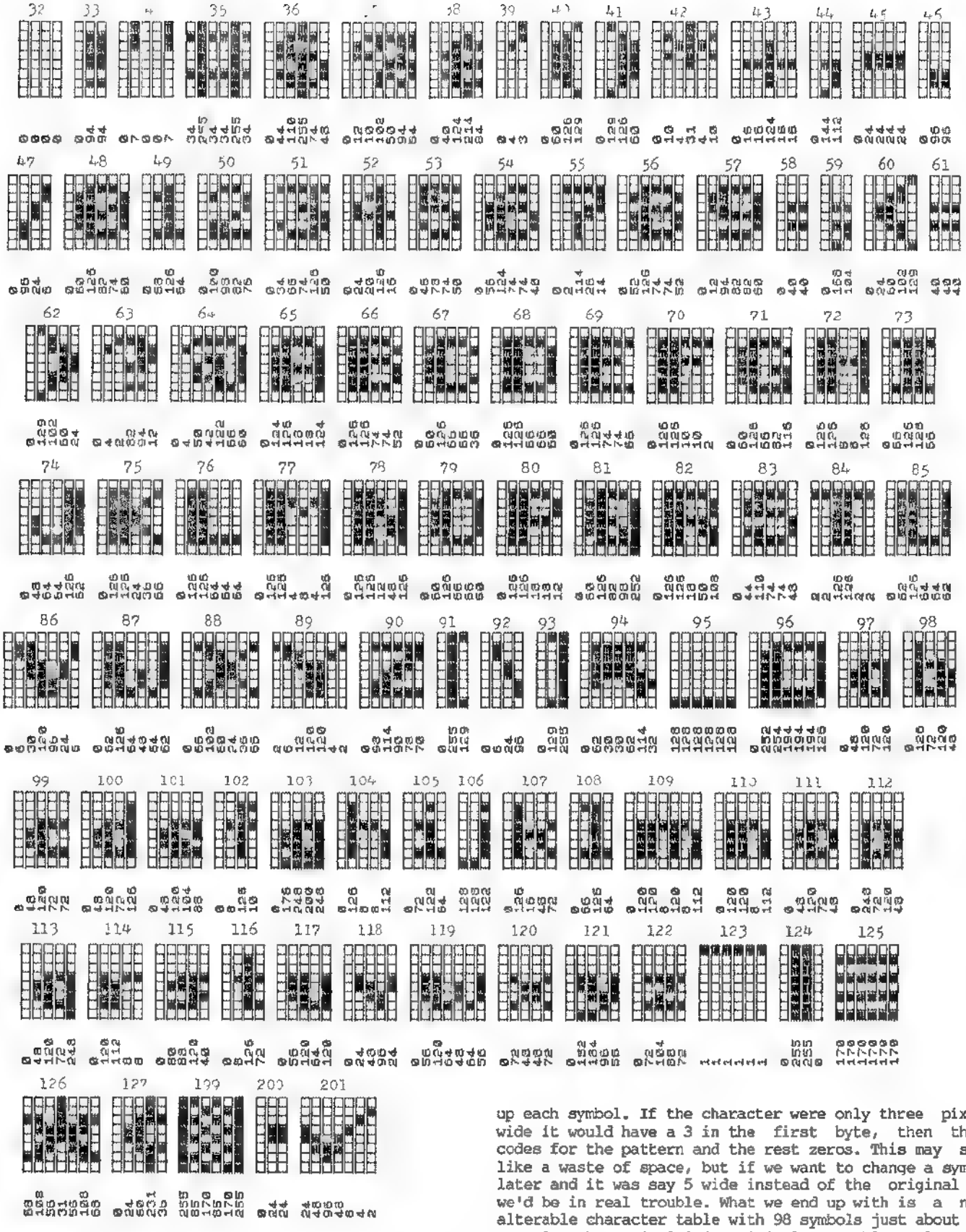
```
 PATERN    CODE   BITS SET
                  00000000
                  00000001
                  00000010
                  00000011
                  00000100
                  00000101
                  00000110
                  00000111
                  00001000

 1 REM   Z80 Bit Paterns
  10 FOR t=65356 TO 65391  READ
a: POKE t,a: NEXT t: DATA 255,12
8,128,128,128,128,255,0,,255,254
,254,254,254,254,255,0,,128,128,
128,128,128,128,128,128,0
  20 DIM b(8): LPRINT AT 0,5;"PA
TERN    CODE   BITS SET"
  30 FOR f=0 TO 255: POKE 16334,
f: LPRINT "          ";: FOR i=1 TO 8
  40 LET b(i)=POINT (i-1,175)
  50 IF b(i)=0 THEN LPRINT CHR$
144;: GO TO 70
  60 LPRINT CHR$ 143;
  70 NEXT i: LPRINT CHR$ 146;"
";f,AT 0,20;b(1),b(2);b(3),b(4),
b(5),b(6),b(7),b(8)   NEXT f
```

But, you might ask, the 2068 uses eight by eight sized symbols. One of CLASSY's functions is to get away from that! Indeed some of CLASSY's symbols are only three pixels wide. However, all of CLASSY's symbols are eight pixels HIGH. By doing a little extra code manipulation using the instructions (covered nicely by Syd Wyncoop in his past articles), we can produce our patterns up instead pf across. This is similar to the method some side-ways printer dump utilities one sees on the market are using.

Examine the complete NEW 2068 MEDIUM font in Figure 1. Note the sequence of bytes for each symbol. The font is nearly identical with the one in Part I with only changes being to characters 126, 199, 200 and 201. The actual data for the font is to be found in Listing 2. Character 32 (a space) is missing as it is only zeros and a simple increment of the printing coordinate will take care of it.

Figure 1



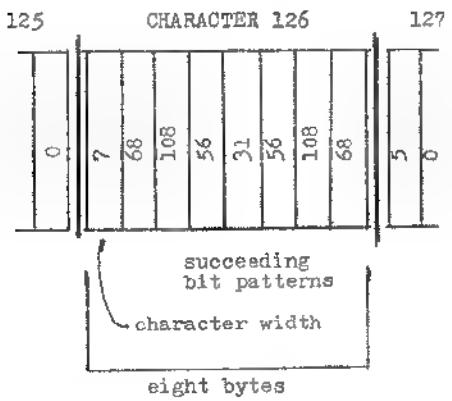
The symbols are each stored, as in the 2068 itself, in eight byte chunks (refer to Figure 2). The first byte tells how wide the character is to be. What follows are the bytes (up to seven as in the case of #126) that make up each symbol. If the character were only three pixels wide it would have a 3 in the first byte, then three codes for the pattern and the rest zeros. This may seem like a waste of space, but if we want to change a symbol later and it was say 5 wide instead of the original 3, we'd be in real trouble. What we end up with is a new, alterable character table with 98 symbols just about the same length as Sinclair's original ROM table. Only this time with a little more pizzaz!

Due to the length of the actual code for CLASSY (it comes in at just over 1.5K), we will be discussing it for the next two installments. Then we will probably get

into "windowing", etc. If you can't wait until next time, I would be happy to send you an advanced copy of the dissassembled listing. Just send one dollar ($1) to: Paul Bingham, P.O. Box 2034, Mesa, AZ 85204. For the program on tape send $5 (for all those who sent for CLASSY from Part I, you need only send $3 for the tape). See you next time!

Figure 2

MEMORY MAP OF FONT TABLE.



Listing 2

```
1 REM  CLASSY FRONT END FONT

1000 DATA 3,0,94,94,0,0,0,0.,5,0
,7,0,0,7,0,0.,6,34,255,34,34,255
,34 0. 6 0,44,110 255,74 48,0
1010 DATA 7,0,12,10,102,50,94,54
. 5,0,40,124,214,84,0,0,3,0,4,3
,0,0 0 0. 4,0,60,126 129,0,0 0.,
4 0,129,126,60,0,0,0
1020 DATA 6,0,10,4,31,4,10.0.,6,
0,16,16,124,16,16,0.,3 0,144,112
,0,0,0,0.,5,0,24,24,24,24,0,0.,3
.0,96,96,0,0,0,0
1030 DATA 4,0,96,24,6 0,0 0.,6 0
 60,126,82,74 60,0. 4 0,66 126,6
4 0 0,0. 5,0,100,98,82 76 0,0. 6
,0,34,66  74,126,58 0
1040 DATA 5,0,24,20 126 16,0,0.
5,0 46,78 74 50,0 0. 6,0 56,124
74 74,48 0.,5,0,2,114 26 14 0,0,
,6 0 52 126, 74,74 52 0
1050 DATA 6,0,12,94 82,82 60 0.,
3 0 40,40 0 0,0 0.,3 0 166,104 0
,0 0 0. 5 0,24,60,102,129.0,0. 3
,40 40,40 0,0,0,0
1060 DATA 5 0,129 102,60 24,0,0.
,6 0,4,2,82 94,12 0. 7,0 4,50 42
 122,66 60. 6,0,124 126 18,18 12
4,0,,6 0 126,126 74  4,52,0
1070 DATA 6 0,60,126 66 66,36 0,
,6 0,126,126,66,66 60,0.,6,0,126
,126,74,74,66,0. 6,0 126,126 10,
10,2,0. 6 0 60,126 66,82,116 0
1080 DATA 6,0,126,126,8,8,126 0,
,5 0,66 126,126,66,0 0. 6,0 48,6
4,64,126,82,0.,6 0,126 126,24,36
 66,0. 6,0 126,126,64,64,64 0
1090 DATA 7,0,126 126,4,8,4 126,
,6,0 126,126,12,48,126,0.,6 0,60
,126 66,66,60,0.,6,0,126,126 18,
18,12,0. 6,0 60,126 82,98 252,0
1100 DATA 6,0 126 126,18,50 108,
0.,6,0,44,110,74,74 48,0.,6,2,2,
126 126,2 2,0.,6,0,62,126,64,64,
62 0. 7 0 6 30,120,96 24,6
1110 DATA 7,0,62,126,64,48,64,62
. 7,0,66,102,60,24,36,66. 7,2,6
12 120,120,4,2.,6,0 98,114,90,78
,70,0. 3,0,255,129,0 0,0,0
1120 DATA 4,0,6,24 96,0,0,0.,3,0
,129,255,0,0,0,0 ,7 0,62 90,30,6
2,114,32.,6,126,128,128,128,128
128,0.,7,0,252,254,194,194,194,1
26
1130 DATA 5,0,48,120 72,120,0,0.
,5,0 126,72,120,48,0 0.,5,0 48,1
20 72,72 0,0.,5,0 48 120,72,126
0 0.,6,0,48,120,104,88,0,0
1140 DATA 4,0,8,126,16,0,0,0.,5
0 176,248,200,248,0,0.,5 0,126,8
,6 112,0 0.,4,0,72,122,84 0,0,0.
,3 128,128,122,0,0,0,0
1150 DATA 5,0,126,16 48,72,0 0.
4,0,66,126 64,0,0,0. 7,0,120,120
 8,120,8,112.,5,0 120 120,8,112
0,0.,5,0,48,120,72,48 0,0
1160 DATA 5,0 248,72 120,48 0.0.
,5,0,48,120,72,248,0 0.,5 0,120
112 8,8 0,0.,5,0 80,88,120,40,0,
0. 4,0,8,126,72,0,0 0
1170 DATA 5 0,56,120,64,120,0,0.
,5,0,24,48,96,24 0 0. 7,0,56,120
,64,46,64,56.,5,,72,48,48,72,0,
0. 5,0,152 184,96,56,0,0
1180 DATA 5,0,72 104,88,72,0,0.,
6 1 1 1,1,1,1 0. 4,0,255,255,0,0
,0,0. 8,170,170,170,170,170,170,
0. 7,68,108,56,31,56,108,68
1190 DATA 5 0,24 60 231,36,0,0.,
6,255 85,170,85,170,255,0.,3 0,2
4,24,0,0,0,0.,7,24,48,96,48 8,4,
2
```

RGB Interface For The TS2068

By

Tim Stoddard

RGB vs Other Methods

Ever since I purchased my TS2068 I have longed for a REAL RGB interface. Sure, the display is OK on my 19" Sony, but the color is "washed out" and there is that ever-present background hash. I thought that by attaching a composite monitor to the monitor jack would at least clean up the hash....but no, the hash remains! Thus began my work on THIS article.

Composite video is simply a combination or mixture of the three basic signals common to all CRT (Cathode Ray Tube) based display devices. First, there is the VERTICAL SYNC which is used to 'lock' the picture vertically. Second, there is the HORIZONTAL SYNC which 'locks' the picture horizontally. Then there is, of course, the VIDEO INFORMATION which supplies the actual picture.

COMMON TVs

In the case of color displays there are three methods of getting these signals to the monitor or TV. In the begining.....there was the RF modulator. This is simply a VERY low power broadcasting station! All three basic signals are combined to from the COMPOSITE signal, and then 'broadcasted' via a cable to your favorite TV. You, in turn, select channel 3 or 4 and, voila! The 'Gulpman' appears on your screen. The advantages to this type of interface are ease of hook-up, cost, and availablity of the common color TV. The disadvantages are the BANDWIDTH of this type of interface are LOW. This is seen as poor picture quality and 'washed out' color. There are two reasons for this; 1) The tuner circuitry in the TV and RF modulator in the TS2068 cannot handle the high resolution modes that our computer is capable of 2) The 'common' TVs' CRT DOT-PITCH is not good enough for high-resolution color images. (More on this later). Then there is that ANNOYING 'background hash'!

COMPOSITE MONITOR

The second method is to send the composite signal directly to the TVs' CRT circuitry, bypassing the tuner and the computers' RF modulator. This method works quite well because we are bypassing all that circuitry. The display looks MUCH better but the color is still 'washed-out' due to the fact that the DOT-PITCH of the TV is large. Also, the circuitry needed to combine the color/sync signals in the computer end, and the circuitry needed to separate the color/sync signals in the monitor end, add to the distortion of the display and color. I probably would not have noticed this except that I also have a QL with its' RGB monitor and there is just no comparing the RGB with a standard monitor or TV. So the Monitor connection is much better than the RF modulator but, still, is not the best interface available.

RGB MONITOR

The third method is, by far, the most impressive in that the quality and control of the color signals are absolute. This is accomplished by sending the three color signals SEPARATELY to the monitor along with a COMPOSITE SYNC signal containing the VERT and HORZ syncs. The disadvantages to this method are the multi-wire cable needed to connect the monitor and the cost of the monitor. Why is the RGB monitor more expensive? Most of the cost of an RGB monitor is tied up in the CRT RESOLUTION and circuitry to support it. A typical TV has a DOT-PITCH (DOT-PITCH is the distance between adjacent CRT pixels) of about .85mm, where a good RGB monitor such as the QL Vision monitor has a dot-pitch of .38mm. You can really see the difference!

So we have two goals here; one, to convert our machines to RGB and two, to get rid of that ANNOYING 'background hash'!

CONSTRUCTION

This circuit was derived from a circuit appearing in the 2068 Technical Manual, page 57 and was modified to work with the QL Vision monitor, although with the proper connector, it should work for any RGB monitor. If you don't have this book, GET IT! This is the single most useful book for the 2068 and is published by TIME DESIGNS MAGAZINE.

We need a VERY CLEAN supply for this circuit to eliminate the background hash. This hash is generated by the computers' SWITCHING REGULATOR. To accomplish this we will be 'double regulating' the supply to the RGB sync circuit. This circuit just extracts the SYNC signals from the TS2068 COMPSITE out-put. The color signals already exsist in the TS2068!

+12 VOLT REGULATOR

Using proper static precautions, remove the top keyboard case and then the internal PCB. Using figure #1, locate and remove the 78L12A regulator. It looks like a garden-variety transistor. Use a solderwick or solder-sucker to remove the solder from the regulators PCB holes and then gently remove the wire leads. Next, install a 78M12 or 7812 regulator in its' place using figure #2 as a guide to orienting the new regulator. NOTE that I'm showing the BACK-SIDE of this regulator UP. Now cut a 12" piece of insulated wire and solder on end to the regulator as shown in figure #2. Leave the other end free for now. Leaving the PCB out of the case and making sure the PCB is insulated from contacting any metal, connect your TV and power up the computer. You should get a NORMAL display with the Sinclair copywrite notice. If you don't POWER OFF!!!! Then check the orientation of the new regulator again and correct. If you have a voltmeter you can also check the output of this regulator to insure that it is at +12 volts. This point is the same point as the wire that was attached to the regulator in figure #2.

Using the schematic in figure #3 assemble the circuit on a small piece of perf board. Then pick an area inside the case to mount the small perf board assembly, (I used the area just above the right hand joystick connector.) Use a small piece of two-sided foam tape to mount the perf board. Next, attach the +12 volt wire previously soldered to the new +12 volt regulator and solder it to the proper point on the perf board (The input of the 78L05 voltage regulator). Now run a small wire from the computer's composite out shown in figure #4 to capacitor C1 on the perf board (see schematic in figure #3). Now run a ground wire from the perf board (emitter of Q2) to a ground point on the computer PCB. A good place is the ground plane just above the right joystick connector. Take the 6-wire shielded cable and solder the wire you'll be using for the COMPOSITE SYNC to the perf board (collector of Q2) Solder the wires you'll be using for the Red, Green, Blue signals to the computer's PCB at the points shown in figure #4, then solder the wire to be used for monitor ground AND the SHIELD wire to the main ground point in the computer which is located in the center of the computer PCB It has a large wire soldered there with many other smaller wires and a cap attached to it. Lastly attach an 8 pin DIN socket to the other end of the 6-wire cable, or if you'll be using a different monitor, the proper plug to match that monitor.

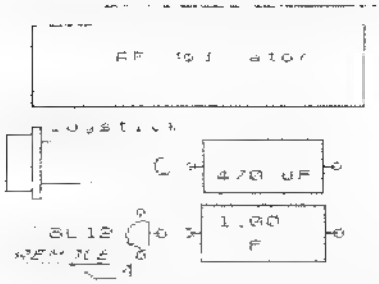


FIGURE #1 +12 Regulator

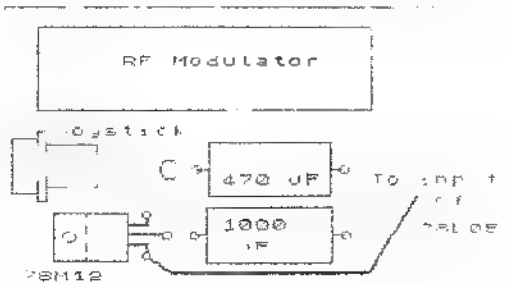
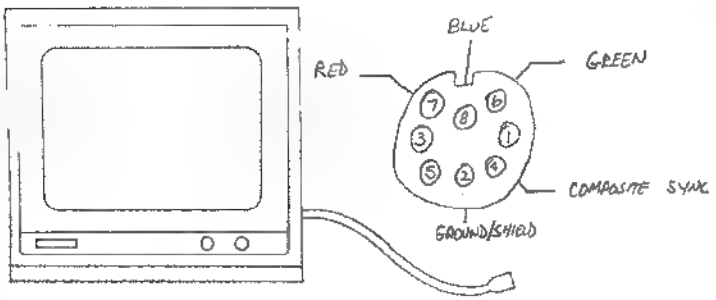


FIGURE #2 New Regulator



QL Monitor Connections

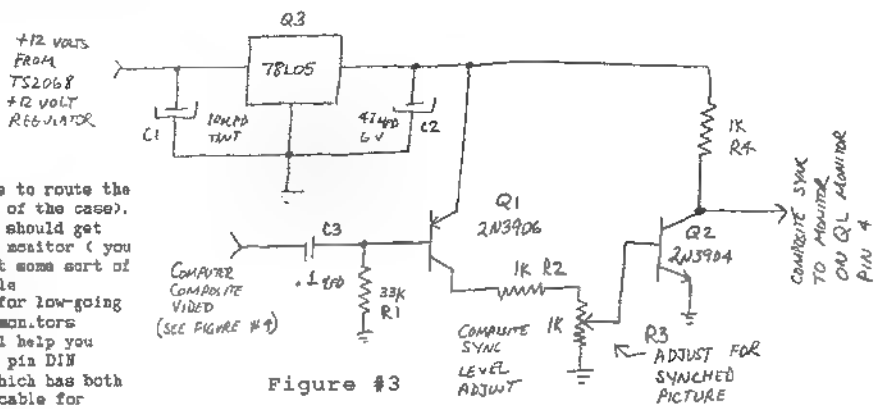


Figure #3

Reassemble the computer into the case and pick a place to route the wire out of the back of the computer (I chose the center-back of the case). Connect a TV and power supply to the TS2068 and power up. You should get the copyright notice as normal Power off and connect the RGB monitor (you can leave the TV connected) and power up again. You should get some sort of display on the RGB Adjust the potentiometer R3 to get a stable display...NOTE NO BACKGROUND HASH!!!This circuit is designed for low-going composite sync monitors which comprises about 90% of the RGB monitors available. If yours is one of the 10%, drop me a line and I'll help you out! All parts are obtainable at Radio Shack, including the 8 pin DIN socket. One source is a Tandy 1000 keyboard extension cable which has both a male and female in-cable connectors plus a 6-wire shielded cable for $14.95 as part number 26-1389. If you want only the female connector they only stock the PCB mountable connector and you'll have to ORDER that as part number AJ7550 for $1 00 (It's a replacement socket for their Tandy 1000 computer!)

LETTERS, NEW BBS, OFFERS!

Thanks for the many letters you have sent in support of my articles! I have improved the TS2016 ram pack upgrade to allow usage with the TS1500. Thanks to reader Earl Donnington of Boynton Beach, Florida, for inspiring me to accomodate the TS1500. For those of you wishing the latest improvements in the upgrades for the ram pack or any other of my articles, just send me a SASE and I ll return the latest info. Also Mark Fendrick, columnist for Computer Shopper has started up a new Sinclair BBS called "SINCLAIR at NIGHT". He has generously named me as the ZX81/TS1000 conference leader, so you can find me daily on that BBS. Please call in on this board and support these great little machines! Hours are 2300 to 0600 daily, phone 718-627-1293. There curently are also TS2068 and QL sub-boards (conferences) with new conferences added to the arrival of new computers such as the THOR. You'll also find any late-breaking info about my articles on that BBS in the ZX81/TS1000 conference. For those of you who are not hardware oriented, I am extending an offer I started with the Vancouver Sinclair User Group to upgrade either the TS2016 ram pack for $25 US, or the TS1000 internal 64K upgrade for $40 US. Just mail the ram pack or computer or both (as many did in VSUG) and I'll convert and return via insured mail. It would be a good idea to check with me ahead of time to see how "swamped" I am. You can do this by mail (85-45 66th Road, Rego Park, NY, 11374), Compuserve (ID 73127,2684), or on the "Sinclair at Night" BBS

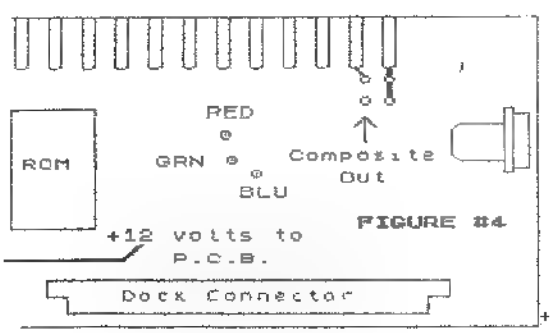


FIGURE #4

Adding Professional Features to TASWORD TWO

By Duncan R. Teague

If you have ever used a word processor for a more expensive computer, like an Apple //e, you have noticed a few features not found on Tasword Two. I regularly use PFS:Write on the Apple //e in my planetarium. (I apologize for having another computer in addition to my beloved TS2068. My only excuse is having to have the Apple to interface with a laser video disc player.)

Both word processors include many identical features: help screens; word wrap; fast cursor movement by character, by word, by line, by page, and to the beginning or end of the text file; character and line insert and delete; block move and copy; left, center, and right line formatting; reformatting of text after insertions or deletions; optional right justification; embedding of printer control codes within the text file; and search and replace. Whew!

Tasword Two is truly amazing, considering that it needs 80K less memory and costs $110.00 less than PFS:Write, but there is room for improvement. There are number of features PFS:Write has that are not included with a stock version of Tasword Two.

The modifications I have made allow Tasword Two to use tape or (Aerco FD68) disk for loads and saves, catalog either of two disk drives, "reboot" either drive, print headers and footers, print multiple copies with all the pages properly numbered, skip perforations after a selectable number of lines, and count the number of words in a text file.

I'm greatly indebted to articles in the February/March 1986 issue of ZX Computing and the May 1985 issue of Sinclair User. Thanks to Mike deSosa for sending me the latter article.

The ZX Computing article was, of course, written about the Spectrum version of Tasword Two. I had to modify the machine code for the word count routine before it would work correctly on the TS2068. This has appeared previously in Time Designs.

The Sinclair User article, by John Lambert, is the earliest source I've seen for selectable lines per page, page numbering, multiple printouts, headers, footers, and custom color selection for the screen. The listing, however, has a number of bugs.

The footer and pagination don't always appear in consistent positions at the bottom of the page. The page count is off under certain conditions. The LPRINT command to print the footer is ignored by the TS2068 version of Tasword Two. Finally, a list of addresses to poke with new values to change the paper and ink colors of the text file, the prompt area at the bottom of the screen, and the margins is incomplete. The information below will "correct" the listing in Sinclair User.

Any modifications to Tasword Two should be preceded by the incorporation of as many memory saving tricks as possible. Mike de Sosa is responsible for several. He used tokens and variables to represent often used GOSUB lines and frequently used numbers for TABs and calculations. Here is a fairly complete list.

0 = NOT PI	12 = tw	64 = sf	4000 = ft
1 = SGN PI	14 = w	850 = ef	6000 = gs
10 = t	31 = th	950 = nf	9000 = nt

One could also have used variables for 0 and 1, say "z" and "o."

Some additional memory was saved by reducing the amount of text in screen prompts and in the menu. I don't believe any clarity is lost by these changes.

Another change I made renders Tasword Two compatible with a word processor whose default page format is a one-inch margin at the top and bottom of the page. The margin area is usually where headers and footers are printed. In keeping with convention, two blank lines are printed, then the header, and then three blank lines. Six lines equals one inch. Then the text file starts. The header is printed on every page in the document, but it uses up no space in the text file.

The footer is likewise separated from the last line of the text file on a given page by three blank lines. The footer is printed next. Then the current page number and total number of pages in the document is printed on the next line. Finally a form feed is sent to the printer. This effectively leaves one blank line after the automatic page numbering.

When I first typed in the Sinclair User listing, the footer would not print out at all. This is why Mike sent me the article in the first place. Repositioning a USR call solved the problem.

The original Sinclair User listing allowed for the number of lines per page to be specified along with Tasword's line spacing choice. It failed to take into account the extra blank lines needed to keep the footer in a consistent place when line spacing is greater than one.

If a document has a one inch margin at the top and bottom, then 54 lines of text will fit on a standard page. If the line spacing is one, then everything gets printed in the right place. If the line spacing is three, then Tasword prints a line and skips two lines until it gets to the 18th line (54/3=18).

It prints the 18th line, then thinks, "I'm through printing lines on that page," falls to put in two blank lines, and prints the footer. This puts the footer and page numbering two lines higher than it should be. An addition to line 290 tells Tasword to print "line spacing minus one" blank lines between the last line of text on the page and the footer and pagination.

Another problem occurred when the Sinclair User listing calculated how many pages it would take to print an entire document, given the user-specified lines per page and the line spacing. If the text just happened to fill the final page of a multi-page document, the page counter calculated one page too many. An addition to line 250 prevents "one" from being added to the page count if the total number of lines divided by the lines per page is an integer.

The final consideration is the alteration of the screen colors. POKE the following addresses to change the text area to your choice of paper and ink, where "C" is calculated by the formula "C = (8 * paper)+ink."

```
POKE 58512, 54        POKE 58521, 54
POKE 58513, C         POKE 58522, C
```

For example, to use red paper with black ink, you use the formula "C = (8 * 2) + 0 = 16."

A surprise may await you after you've changed the text area paper and ink colors. If you try to set left and right margins, your text may disappear or your margins may be invisible if you chose certain incompatible combinations. To alter the paper and ink colors in the margins for suitable contrast and visibility, POKE the following addresses with your selection according to the aforementioned formula.

```
POKE 58508, 54        POKE 58517, 54
POKE 58509, C         POKE 58518, C
```

Now that you've customized the text area and margins, you can alter the ink and paper of the bottom two lines where line and column numbers, justification, word wrap, and insert mode status are shown. To alter the two status lines, POKE new values into the respective addresses. The value of "C" is calculated in the manner shown previously.

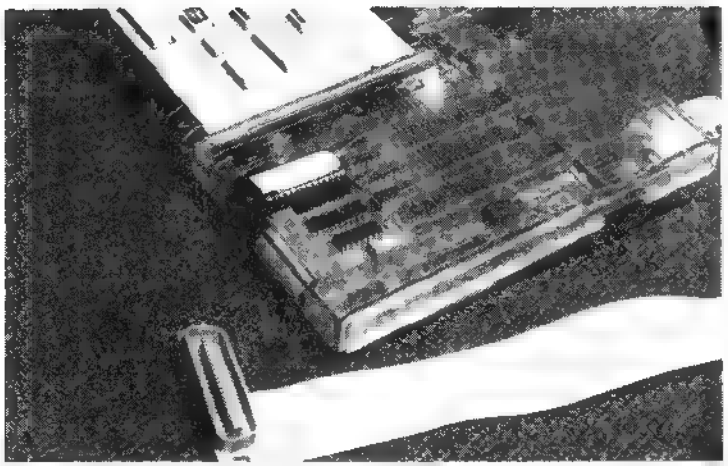
```
POKE 59993, C (top)  POKE 64570, C (bottom)
```

The final touch-up is color selection for the 64-column and 32-column mode borders. You can change the following addresses to indicate your choice of BORDER color. The value of "B" equals the paper color you want. Thus "B" ranges from 0 to 7.

```
POKE 64516, B (64-column)   POKE 60641, B (32-column)
```

Continued on Page 20...

FOOTE SOFTWARE



The FOOTE PRINT PRINTER INTERFACE

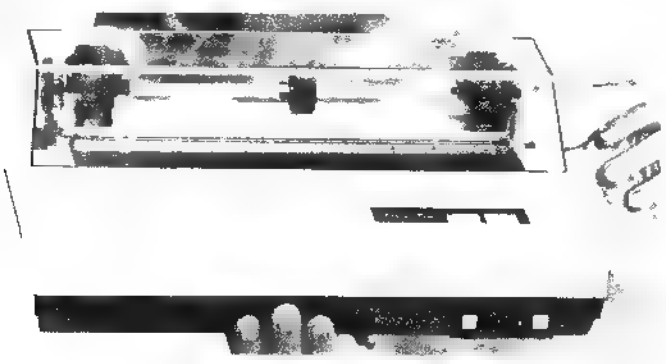
- for Centronics parallel printers
- works in both 2068 and Spectrum mode
- compatible with OS-64 & Spectrum emulators
- EPROM socket and on/off switch on board
- works with **both** Tasman and Aerco driver software
- plugs into cartridge dock—door completely closes with cable running back under computer
- frees up rear edge connector allowing other peripherals to be used; less chance of a crash
- print driver software for LPRINT, LLIST, and COPY included for 2068 and Spectrum modes

FootePrint Interface w/software & cable .$45.00
FootePrint with OS-64 option included ..$65.00
Bare board & instructions only $20.00
Cable only for use with bare board .$15.00

Zero Insertion Force Socket option add $10

SOFTWARE	TS2068	TS1000
Badgammon (Backgammon)	$12.95	
Advanced Math (Calculus)	$12.95	$7.95
Calorie Counter	$9.95	$5.95
U.S.A. (Pres. & States & Caps.)	$9.95	$5.95
Gambler (poker)	$9.95	
CHR$ (char. & graphics generator)	$12.95	
Hangman & TIC–TAC–TOE	$5.95	

Brother M1109 Dot Matrix Printer, compact, low noise, 100 CPS, both Parallel and Serial interfaces, multiple typestyles with near letter quality print mode and 4k memory buffer, comes with tractor feed unit..... **$249.95**
QL or Zebra FDD cable for above: **$17.00**



The Best of SUM

Some sample articles include. Building Your Own Spectrum Emulator, Repairing Your TS-1000, Word Processing Reviews for the 2068, UDGs on the TS-1000, Extensive Review of the Zebra Disk System, Adding a Keyboard to the 2068, and Enhancing the A & J Microdrive 112 pages

Price: $11.95

THE BEST OF

S TS 1000 **U** TS 1500 **M** TS 2068

the SINCLAIR/TIMEX USERS MAGAZINE

A compilation of the best articles, reviews, programs, and hardware projects

The Best of SUM, Part II

Articles include Building an EPROM Programmer Sprites on the 2068 Adding RGB to 2068 QL Word Processing, What's Available for TS-1000 and much more 60 pages

Price: $7.95

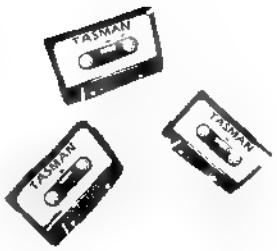
FOOTE SOFTWARE P. O. Box 14655 — Gainesville, FL 32604 904/462-1086 (6 pm - 9 pm EDT)
All prices are pre-paid and include shipping charges. Florida residents must add 5% state sales tax.

The final modifications I've made to my copy of Tasword involve the use of my Aerco FD68 Disk system. The save and load routines allow both tape and disk to be used. The difference is in the use of an "*" to prefix disk file names or to catalog the disk before loading or merging a text file. At Tasword's "STOP" menu, you can also use appropriate keys to catalog drive A (<=), catalog drive B (>=), or reboot (<>).

If you'd like to save yourself some typing, I'll supply a tape or disk of my modifications to the BASIC portion of Tasword for $6.00 postpaid. I'll also include the machine code routine which provides a count of the number of words in the text file when you go to the main menu. Contact me at 3388 Bluemont Drive, Memphis, Tennessee (38134).

Here are some notes on the program lines:

```
15 - 20: variables defined for memory savings
25       : my choice of STOP menu paper and ink colors
         : USR 52610 is for the word count routine
95       : establishes "<=" key to catalog drive A
100      : establishes ">=" key to catalog drive B
105      : establishes "<>" key to reboot selected drive
200      : lines/page POKEd into unused address 33220
225      : "len" = total characters to print
250      : "pl" = total characters to print on each page
         : "tp" = total pages to print (if len/pl is not
           an integer, then "1" is added to "tp")
265      : "a$" is temporarily the document's header
270      : 2 line feeds; the header; 3 line feeds
290      : proper # blank lines print if line spacing >1
292      : "a$" is temporarily the document's footer
294      : 3 line feeds; then the footer;
296      : then the pagination; then a form feed
298      : USR 59806 moved here to allow "LPRINTs" above
299      : "f" is pages; "n" is number of copies
700      : save BASIC ("boot" name for disk auto boot)
705      : save machine code (extra code for word count)
1000 ff: tape/disk text file save routines
2000 ff: tape/disk text file load/merge routines
8000     : GOSUB to center the header and footer
9000     : GOSUB frequently used by print options
```




```
  10 CLS   LET a=USR VAL "64330"
  GO TO t+1
  11 BEEP VAL ".2",VAL "3"  BEEP
VAL ".3",VAL "2"  RETURN
  15 POKE VAL "23609",VAL "2"
CLEAR VAL "33279"   LET t=VAL "10"
   LET sf=VAL "64"   LET ft=VAL "4
000"   GO SUB ft  CAT "tasword.bi
n"   CLS   GO SUB VAL "11"  LET
a=USR VAL "59081"   GO TO t
  20 CLS   LET a=sf*INT (a/sf+VA
L "0.99")   LET tw=VAL "32"   LET
w=VAL "14"   LET th=VAL "31"   LET
 ef=VAL "850"   LET nf=VAL "950"
  LET gs=VAL "6000"   LET nt=VAL "
9000"   IF a<>NOT PI THEN GO TO nt
*gs
  25 PAPER VAL "5"   INK NOT PI
CLS   GO SUB ft  PRINT AT VAL "4
",NOT PI,"Edit file [",USR VAL
52610","words]" TAB th,"Y"
  28 PRINT ''Save file",TAB th,
S
  30 PRINT ''Load file",TAB th,"
J
  35 PRINT   Merge file",TAB th,
"M"
  40 PRINT ''Print file",TAB th,
"P"
  45 PRINT ''Save Tasword FD68",
TAB th,"T"
  50 PRINT ''Graphics/Printer",T
AB th,"G"
  55 PRINT ''BASIC",TAB th,"B"
  70 PRINT AT t+3,VAL "11";"Pres
s key"
  80 LET a$=INKEY$   IF a$="" THE
N GO TO VAL "60"
  90 LET a=NOT PI   LET b=CODE a$
   IF b<VAL "97" THEN LET b=b+th+
SGN PI
  95 IF b=VAL "199" THEN CAT "a
",   GO TO VAL "720"
 100 IF b=VAL "200" THEN CAT "b
",   GO TO VAL "720"
 105 IF b=VAL "201" THEN CAT "bo
ot,bas"
 110 IF b=VAL "115" THEN LET a=V
AL "6"
 120 IF b=VAL "106" THEN LET a=V
AL "8"
 125 IF b=VAL "118" THEN LET a=w
 130 IF b=VAL "112" THEN LET a=t
w
 140 IF b=VAL "121" THEN LET a=V
AL "4"
 150 IF b=VAL "109" THEN LET a=t
 160 IF b=VAL  103  THEN LET a=V
AL  16
 170 IF b=VAL "98" THEN LET a=VA
L "18"
 180 IF a<>NOT PI THEN PRINT AT a
,th; FLASH SGN PI;CHR$ (b-th-SGN
 PI);   GO TO VAL "500"
 190 GO TO VAL "80"
 200 CLS   GO SUB ft  PRINT AT V
AL "2",VAL "9";"PRINT OPTIONS"
PRINT 'ENTER for defaults"  LET
a=VAL "6"  LET a=NOT PI   LET a=0=
t+tw  LET x=VAL "33220"  LET a$=
"Lines/Page=   GO SUB e'  LET a=
VAL "3"  PRINT AT a,NOT PI;"Line
 Spacing=1   GO SUB gs   GO SUB n
t
 210 POKE VAL "62235",VAL a$   LE
T a=t   PRINT AT a,NOT PI;"Start
 line=1"   GO SUB gs   GO SUB nt
 220 LET c=sf*(INT VAL a$ SGN PI
)   LET start=FN p(VAL "62216")   L
ET a=tw  PRINT AT a,NOT PI;"Fini
sh line=last"   GO SUB gs   IF a$=
```

```
 700 LET a=VAL "8"   MOVE "boot.b
a$",15
 705 MOVE "tasword.b.n",52610,12
9
 710 CLS   CAT ''
 720 PRINT #NOT PI,AT NOT PI,NOT
 PI;"Press ENTER to go back to
menu"   PAUSE NOT PI   GO TO VAL
"8"
1000 LET b=FN p(VAL "62216")   CL
S
1005 PRINT AT VAL "8",NOT PI,"Na
me text file-Prefix * for disk"
   LET a=t   LET a0=NOT PI   GO SUB
gs
1010 IF LEN a$>1 AND a$(SGN PI)<
>"*" THEN CLS   PRINT AT tw,NOT
PI;"Too many characters - max is
 10"   GO TO VAL "1005"
1020 IF LEN a$=NOT PI THEN   CLS
   PRINT AT tw,NOT PI;"There must
 be a name"   GO TO VAL "1005"
1025 IF a$(SGN PI)="*" THEN LET
a$=a$(VAL "2" TO ): LET a=tw   LE
T a$=a$+" bin","+STR$ b+","+STR
$ a   MOVE "a$",,, CLS   GO TO VAL
 "1040"
1030 LET a=tw: SAVE a$CODE b,a
CLS
1040 PRINT AT VAL "8",NOT PI,a$,
" saved"  AT t,NOT PI;a;" bytes
"   a/PEEK VAL "62237";" lines"
1050 PRINT AT tw,NOT PI;"Verify
";a$,"";   LET a=w   GO SUB VAL "9
20": IF a<>NOT PI THEN GO TO t+t
1060 IF LEN a$>1 THEN GO TO VAL
"710"
1100 CLS   PRINT "Start tape"   V
ERIFY a$CODE b,a
1110 PRINT AT th-t,VAL "8";"text
 file verified"   GO TO VAL "25"
2000 CLS   PRINT AT VAL "8",NOT
PI,"Type the name of the text fi
le",AT t,NOT PI;"and press ENTER
"
2010 PRINT AT tw,VAL "2";"Just p
ress ENTER to load the",AT w,VAL
 "2";"first tape file, or * to C
AT"
2015 LET a0=NOT PI   LET a=VAL "1
6"   GO SUB gs
2020 IF a$<>"*" THEN PRINT AT VA
L "18",VAL "9","Play the tape"
GO TO VAL "2030"
2025 CLS   CAT '',   INPUT "Type
filename",a$   LET a$=a$+".bin""
"
2026 IF b=109 THEN LET b=FN p(VA
L "62216")   LET c=((FN p(VAL "62
221"+VAL "22")*sf+a)   LET a$=a$
+STR$ (a+b)+","+STR$ c
2029 CAT a$;   GO TO t
2030 LET b=FN p(VAL "62216")   LO
AD a$CODE (a+b),((FN p(VAL "6222
1")+t+tw)*sf-a)   GO TO t


8000 FOR g=SGN PI TO (VAL "80"-L
EN a$)/VAL "2":  LPRINT " ";   NEX
T g:  LPRINT a$;CHR$ t   RETURN
9000 IF a$="" THEN LET a$="1"
9010 RETURN
```

Zebra Systems, Inc.

78-06 Jamaica Ave., Woodhaven NY 11421 (718) 296-2385

News from Zebra Systems

Zebra Systems has excitement buzzing in the air! We've added new staff, new mailist capabilities and, our very own desktop publishing group to keep our customers up to date on Timex Sinclair product changes, new product introductions, industry happenings, and more!

The recently released Summer Issue of the Zebra Systems catalog is our very first desktop publishing achievement, and we are very proud. It is attached here for your shopping convenience, and we know you are going to like the products and the prices.

We Need Your Help To Send You Free Catalogs!

During the first week of July, Zebra Systems mailed out 8,000 free catalogs to our recent customers, those of you who requested catalogs, and T/S Clubs. We are well aware that our mailist has not previously been kept up to date with all your current mailing address information. For this reason, we are asking all Timex Sinclair Users who wish to remain on our mailing list and continue to receive our free Timex Sinclair Products Catalogs to return the coupon on this page or, the back cover of the Summer Catalog just mailed to you. Don't delay. We have already begun work on our Fall 1987 Catalog.

Attention Timex-Sinclair Club Members

We will gladly send you as many catalogs as you wish for your club meetings or for inclusion in your club mailings. Or, if you prefer, send us your club membership list, and we will mail catalogs directly to each member. In addition, our customers often ask us if we know of an active TS Club in their part of the country that they can participate in. If you have any doubts that we know about your club's existence, please write to us.

FDD User Newsletters Mailed

Zebra Systems recently mailed the first in a series of informative newsletters to all of the registered Zebra FDD Users. (If you own a Zebra FDD and did not get your copy, be sure to contact us immediately.) The newsletter received tremendous applause from those active FDD supporters.

MTERM Price Drop!

As a correction to our Summer Catalog we want to announce that the prices on both MTERM/T and MTERM II have been drastically reduced. These are the prices we intended to have on catalog page 9: MTERM/T is $8.95, and MTERM/II is $14.95.

Our Thanks

We would like to thank Frank Davis, Paul Holmgren, and all of the other people who worked so hard to help make the 1987 MidWest Timex Sinclair Computer Festival such a great success.

✔ **Yes, I want to be on the Zebra Mailist.**

Name_____

Address_____

City_____

State____Zip_____

Computer:

__ZX81/TS1000

__TS1500

__TS2068

__QL

__TS2040 or Alphacom 32.

80-Col. Printer:_____

Printer Interface_____

Modem_____

Do you ever call BBS's?___

Computer Club Membership:_____

Comments:_____

_____

_____

REVIEWS

Larken Disk-Operating System

DAVID SOLLY

When I was looking for a disk operating system, the first thing I wanted was a system that would work on ALL of the ROM configurations available for the TS2068. At that time, I was running MSCRIPT, TASWORD II with a Spectrum Emulator, and ZTERM modem software with the Zebra OS-64 cartridge. Secondly, I was looking for a system which would leave me the largest amount of RAM possible because I was working mostly with large data-base and word processor type programs. Finally, I needed a system that would be simple to use and also within my limited budget. I searched in vain until the answer appeared almost in my own back yard [Ed.- Mr. Solly currently lives in Ottawa, Ontario, Canada].

The new Larken DOS dveloped by Larry Kenny of Larken Electronics, was like having my every wish come true. The disk operating system (DOS) fits neatly onto a cartridge which loads into the Timex Command Cartridge port. For a little bit extra, a second socket, fitted with a switch, can be added to the cartridge, to hold an additional chip that the user may own, such as a Spectrum Emulator, or a Zebra OS-64.

The rear circuit card is a compact double-density disk drive interface which also has a Kempston joystick port and a Non-Maskable Interrupt (NMI) push button. Because the DOS is "burned into its own EPROM, all the RAM of the computer is available to the user. The DOS is completely accessable whether you are using the standard Timex Sinclair ROM, the Spectrum Emulator ROM, or the Zebra OS-64 software ROM. Programs which are de-signed to run on both the TS2068 and Spectrum, need not be saved in two different formats to be re-loaded into each ROM configuration. A formated double-sided, double-density 80-track disk drive is capable of saving up to 800K of programs.

Converting programs to run on the Larken DOS, once the hardware is installed and the DOS is called into operation by RANDOMIZE USR 100: OPEN# 4, "dd", is essentially a snap. The Larken DOS uses the same Sinclair tokens (which are found on the 2068 keyboard), as the tape-driven systems. After learning a few simple instructions and four mnemonic extensions, I was ready to handle most programs. The necessary adjustments needed to convert tape drive commands to Larken DOS commands can be quickly illustrated by the following chart of equivalents.

Tape Drive Command	Larken DOS Command
SAVE "Name"	PRINT #4; SAVE "Name.Bn"
SAVE "Name" LINE 10	PRINT #4; SAVE "Name.Bn" LINE 10
SAVE "Name" CODE	[No equivalent]
SAVE "Name" CODE s,l	PRINT #4; SAVE "Name.Cn" CODE s,l
SAVE "Name" SCREEN$	PRINT #4; SAVE "Name.Cn" SCREEN$
SAVE "Name" DATA A()	PRINT #4; SAVE "Name.An" DATA A()
SAVE "Name" DATA A$()	PRINT #4; SAVE "Name.A$" DATA A$()
LOAD "Name"	PRINT #4; LOAD "Name.Bn"
LOAD "Name" LINE 10	PRINT #4; LOAD "Name.Bn" LINE 10
LOAD "Name" CODE	PRINT #4; LOAD "Name.Cn" CODE
LOAD "Name" CODE s	PRINT #4; LOAD "Name.Cn" CODE s
LOAD "Name" CODE s,l	PRINT #4; LOAD "Name.Cn" CODE s,l
LOAD "Name" SCREEN$	PRINT #4; LOAD "Name.Cn" SCREEN$
LOAD "Name" DATA A()	PRINT #4; LOAD "Name.An" DATA A()
LOAD "Name" DATA A$()	PRINT #4; LOAD "Name.A$" DATA A$()

A "name" in LKDOS can be any combination of letters, numbers and graphics, up to six characters plus the appropriate extension. The "name" may also be in an alpha-numeric string so long as it follows the proper format and its total length is not longer than nine characters. If the "name" is to be stored in a DIM'ed alpha-numeric array, which I do not recommend although it can be done, then the array can not be any longer than nine characters of which the final three characters must be the proper extension.

The Larken DOS also has commands and error reports which are specific to its own operating system plus a number of extended BASIC commands. Here are two examples of Larken commands:

PRINT #4: CAT "",

This is a very useful command that not only allows the user to see what has been stored on the disk but also, thanks to the use of unique extensions, what kind of files have been stored, the length in blocks of each file, and the total number of free blocks available. In the TS2068 ROM and the Zebra OS-64 mode, the user can also specifiy which type of file he wishes to be listed by entering, for example, CAT ".B", when searching for just BASIC programs. Specific program names can also be searched in this manner.

PRINT #4: OPEN# 3,"lp"

This command will allow you to LPRINT and LLIST to a full size printer that uses Aerco-type printer in-terface.

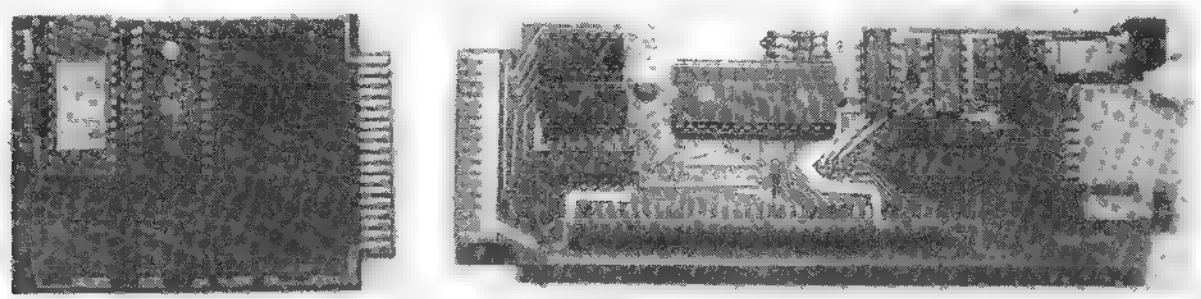
The extended BASIC commands available on Larken DOS allow the user to create geometric shapes with the DRAW and CIRCLE commands and then to fill them with any of 10 different patterns. Also available are a double POKE command (also known in some BASIC's as DOKE), window creating commands, and a command which will read and LPRINT an ASCII text file directly from disk.

Finally, the Larken DOS has a number of new error reports to make operations easier for the user. These reports are:

S - Protect Error
T - File Not Found
U - Disk Full
V - Wrong File Type
W - Invalid Command
X - Cat Data Error
- CRC Error (number)

Larry Kenny told me that he plans to add sequential file capabilities and other disk support software to his Larken system.

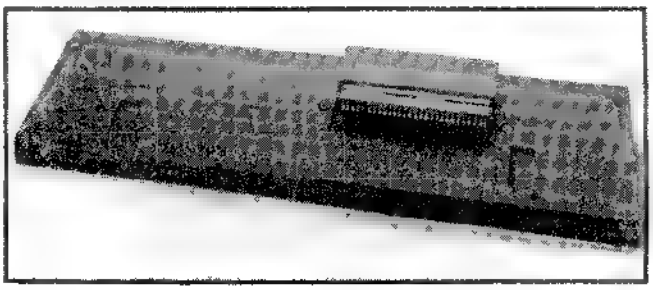
The cost of the new Larken system for the TS2068 (which does not include the actual drives or disk drive power supply), is $119.95 (U.S.). Optional extra socket for add-on ROMs is $6.00. The system may be ordered directly from the manufacturer: Larken Electronics, RR #2 Navan, Ontario, Canada K4B-1H9, tel. (613) 835-2680; or several dealers have them in stock, including RMG Enterprises, Weymil Corp., and Variety Sales.



THE NEW LARKEN DISK-OPERATING SYSTEM: consists of two parts a) a cartridge board that fits into the Timex Sinclair 2068 cartridge port which contains the DOS software, and b) a circuit card that plugs into the rear expansion slot, which is the actual disk drive interface. This card also has a feed-through connector for attaching other Timex peripherals.

The Disciple Disk-Operating System

TONY BROOKS



The Disciple Interface is primarily a disk interface, but it also offers several other features which probably make it the best value in disk interfaces for the Sinclair Spectrum currently available. The Disciple is intended for use on a 48K Spectrum. To use it on a TS2068, it requires the addition of a "twister" board and a Spectrum ROM or Emulator. I have found that the Disciple works with every combination of Spectrum ROM, ROMSWITCH, Emulator, and twistor board I have tried. I even had success with the TK90X ROM from the Brazilian clone of the Spectrum. In this respect, the Disciple is much more tolerant than the Sinclair Microdrives.

The Disciple will support any of the popular size drives from 3 inches to 5.25 inches, in single, double or quad density. Up to two drives can be supported and they may have different specs. Thus drives of different sizes and densities can be used at the same time.

The DOS is supplied on a cassette tape. When I first received my Disciple, DOS version 2b was supplied but turned out to be a very poor quality recording which took much effort to LOAD. This experience was shared by users who received this DOS version. I wrote to Rockfort Products about this tape, and promptly received a new tape with DOS version 2c on it. This new tape loaded just fine, and the new DOS version also solved one other problem, namely, the earlier DOS was prone to resetting the computer from time to time...the new version 2c DOS does not have this problem.

The advantage of having the DOS system on tape is that you can easily configure it to meet your own requirements. I think this is an improvement over those disk systems which have their DOS on EPROM, that is not so easy to change. Upon loading the system tape, you are asked a series of questions to configure the DOS to your disk set up and also to configure the built-in printer interface and network port. When this is done, you are asked if you wish to format a disc and save the configured DOS to that disk. Thereafter, the Disciple may be started up by inserting the system disk in drive 1 and hitting the RUN key on the 2068. Loading of the DOS and programs is very fast--a maximum of 3.5 seconds for a full 48K program. Thus from switching on the computer, loading the DOS and then a program takes less than 10 seconds. The DOS is resident in the Disciples memory and so takes up none of the memory (RAM) of the TS2068 or Spectrum.

I put a copy of the DOS on every disk since it only takes up 3K of disk space. The amount of space on a disk can be up to 720K after formatting. This is enough for fourteen 48K programs on a double-sided 80 track disk if one allows some space for the DOS. One program can have the title "autoload" and this will automatically load after the DOS. I put a one line autoload program to CATalog the contents of a disk on each disk I use.

I am using the Disciple with a single-sided 40 track 5.25 inch drive left over from an unfortunate experience with the Kempston Disk Interface. I have turned most of my disks into "flippies" by cutting an extra notch in them so I can use both sides on a single sided drive. This means I have had the DOS on both sides of the disk but I can get a minimum of six 48K programs per disk.

The DOS uses two kinds of syntax. One is an exact clone of the Sinclair Microdrive syntax, the other is a "simplified" syntax. Both syntaxes may be mixed freely. The simplified Disciple syntax usually consists of placing "d1" or "d2" between the instruction and the filename. Thus loading a Disciple disc file can be achieved by using the command LOAD d1 "filename", and formatting a disk with the command FORMAT d1. Programs can also be loaded by typing LOAD pn--where n is the file number given when you CATalog a disk. This CATalog function gives details of file type, and for code files their size and location. The microdrive syntax is very useful when transferring microdrive based software to disk since usually these programs will run from disk without any alteration. I was able to transfer microdrive versions of TASWORD II and THE LAST WORD to disk and use them immediately without any change.

No disk system is of much use if you cannot transfer your software to disk fairly easily. The Disciple has one of those NMI "magic buttons" that are so popular on many mass storage interfaces. The Disciple magic button works as well as any, taking just a few seconds to save a program from ANY point. Such files takes up a full 48K of disk space regardless of file length. Since the Disciple takes up no 2068 memory it is also easy to transfer software by breaking it down into its component parts, but unless you feel a need to conserve disk space or like to analyize software, it is even easier to use the NMI button. 5.25 disks are so cheap that I don't feel much need to conserve disk space.

The Disciple printer port is a very tolerant beast. The microdrive versions of TASWORD II and THE LAST WORD I referred to above, both printed just fine, even though the former word processor was set up for a TASMAN B printer interface, and the latter program for a Euro-electronics ZXLPRINT III interface. It is also possible to dump a screen to a printer by simultaneously pressing CAPS SHIFT and the Disciple "magic button".

The Disciple has two standard Atari type joystick ports. The right hand port is said to be compatible with Sinclair or Kempston protocol--the left hand port is only Sinclair compatible. However, I have had problems getting the joysticks to operate properly using the Sinclair protocol. The right port has worked fine for me in Kempston mode.

I have not tried using the network facility very much. However, I have been able to send software from a 2068 fitted with Sinclair microdrives to the Disciple network port, and vice versa. I have also successfully sent programs from the Disciple to a QL.

Does the Disciple have any drawbacks? So far only one thing has disappointed me. I have been unable to operate the Disciple with any peripheral attached. This is not too serious since the Disciple has effectively replaced my separate joystick adaptor, printer interface, Interface 1, and Multiface 1. However, I would like to have been able to use Multiface 1 and a light pen interface with the Disciple. The Disciple does have an inhibit button which disables it and then one can use other peripherals, however this does prevent one from accessing software from disc which rather defeats the purpose of the disciple.

Overall the Disciple must be considered a good value with the cost at around $120.00 at current exchange rates. I ordered mine direct from Rockfort Products at 81 Church Road, London NW4 4DP, United Kingdom.

AMX Mouse

MIKE FELERSKI

From the time I first got my hands on the Zebra Graphics Tablet, I have been investigating the use of a mouse for my TS2068. Granted, the Zebra Track ball for the 2068 is quite impressive, but my search for a Mouse continued. After a little British research I found that the best choices for Spectrum mice are the Kempston and the AMX.

My choice was the AMX Mouse by Advanced Memory Systems mainly because it was available here in the states from Curry Computer, who highly recomended it. The Mouse Interface plugs into any Spectrum expansion port, such as the Rainbow Interface, and it operates in the Spectrum mode. The back of the Mouse adapter contains two sockets, one for the mouse itself, and the other is for a Centronics parallel printer cable (the Tasman cable works just fine.)

The software loads in several parts. The first part asks whether or not you wish to save to microdrive. If you answer no then the software skips over loading the second part and goes on to load the third part which is the AMX ART program. As soon as the program has loaded, you are presented with boreded work area surrounded by Icons on the right and pul down menu headers above. In the center of the screen is a hand Icon which moves with the movement of the mouse. This pointer is used to select any of the Icons or the menu headers by just pointing to the desired function and pressing the leftmost button on the mouse.

The AMX ART program operates the same way as Mac Draw or Mac Paint on the MacIntosh. You can draw circles, boxes, and lines. You may draw using a pencil or a spray can with any number of patterns and shades. Imput/Output includes the TS2040 printer, the Interface 1 serial printer port or the AMX printer port with the proper cable. Using the TS2040 printer you are only able to print the immediate screen area, but with a full size printer you are able to select the area you wish to print which includes a larger area than the screen.

In addition to the AMX ART program, there is also the AMX Colour Palette program which allows you to add color to your drawings. The CP program is disappointing only in that it only allows you to add color in large blocks. On the other side of the tape are the AMX control and Icon Designer programs which allow you to create your own mouse software. These I will discuss in the next article.

Over all I have been very impressed with the AMX Mouse. It provides an excellent mouse/Icon enviroment at a reasonable cost ($100 or less.) The only thing lacking in the system is a more detailed owners manual. In future articles I plan to discuss writing software for the mouse, and other software available for the AMX Mouse from other companies.

Nova 1000

Syd Wyncoop

Nova 1000 claims to provide the humble TS1000 with the ability to perform multi-tasking. In case you are unaware, the usual method to achieve multi-tasking is to use one of the newer CPU's The reason is that they have the additional registers and have been specifically designed for multi-tasking. Obviously then, this is no small claim for the Z80 CPU or the TS1000.

The tape comes with three versions of the program on it. The first version contains the machine code in an uneditable line 0, to allow you to add your program lines. This version also contains an impressive demonstration routine. It would have been a nice touch to have provided the 0 REM line by itself. As the program is delivered, you must delete all the unwanted lines.

Much of the how to use this program information must be gleaned from the demo as the two page documentation is not very informative.

Version 2 is an auto-relocatable version, you supply the address and it moves itself. Version 3 will load and move itself above Ramtop. All three versions require 500 bytes overhead with the loader.

The program boasts a real-time clock however, it is a 99 hour clock and I would not find it of much use. The clock readout is in the 99:99:99:99 format and appears in the upper left corner of the screen. The clock requires its own string variable for communication with the user program.

There is also a trace function which shows the Basic line being executed in the upper right corner of the screen. I like the idea but found it to be of little value as presented

Nova 1000 claims up to 400% speed increases over the normal TS Basic. The time savings are obtained by manipulation of the display file size and I could not verify them.

The last feature is an auto-repeat on the keyboard. This I liked and found to be the most useful. It really speeds up keyboard entry

All of these functions are controlled by poking various values into the program (to set program variables) before calling Nova 1000. Nova is then called with a variety of RAND USR calls, depending on the function desired. Nova 1000 will require a new programming discipline. Briefly, you need to set variables to pass parameters to Nova, then make the RAND USR call with perhaps another parameter behind it. I think most beginning programmers will be baffled by Nova 1000. The reason is that the program listing will not be very clear, due to the many and varied USR calls needed to make Nova 1000 function.

Nova 1000 appears to be pseudo-multi-tasking I have not had time to experiment enough yet so I cannot be sure what is happening It appears that Nova is capable of running a Basic and a MC program concurrently, but not two Basic programs Nova does therefore provide a multi-tasking environment, albeit limited

My copy came from Weymil Corp. via RMG Enterprises. The last advertised price I saw was $20 00 and the ad was as informative as the documentation However, better documentation, some good clear examples, and Nova 1000 could be a winner

Beginning Z80 Machine Code

LESSON NINE

BY SYD WYNCOOP

It has been pointed out to me, by an astute reader, that I neglected to tell you to run your MC routines in SLOW, if you are using the TS1000. Otherwise, you cannot see the display of any of my examples. Sorry about that.

This time we will discuss the I/O instructions. For those of you that are wondering what I/O means, it is Input and Output. When I was new to computerdom, I thought I/O referred to my financial status.

To what are we Inputing and Outputing? The computer, but it is actually our old friend, CPU. The I/O instructions allow the CPU to receive or send information through the concept of PORTS and they accomplish this depending upon how the manufacturer made the hardware surrounding the CPU. For example, in our computers port FEh is used for the keyboard. There are others used by Sinclair for the 2040 printer, cassette, and on the 2068, for the bank switching and video mode changes. These are 'hard-wired' in the computer and supported by the operating system, therefore we cannot change them.

What is a PORT? Very simply, it is the doorway though which information flows to and from the CPU and outside devices. There are 256 ports available to us on the Z80 (there are really 65,536, but we will not consider them here). The ports are of course numbered 0-255, as they must be referred to in a single byte. Think of each port as a door to a storeroom. Each door has a number on it, much like a motel would. Each storeroom can hold one byte of data at a time. The CPU can put data in or take data out, by referring to each port (door).

The I/O instructions are In and Out, respectively and there are two forms of the instructions, as detailed in the syntax chart. We are looking at some instructions that are almost english and fairly easily understood.

The forms In A,(n) and Out (n),A use the port specified by n and reads (In) data into A or writes (Out) data from A. This is very similar to the Basic In and Out commands, except that the data is stored in the accumulator. None of the flags are affected by these instructions.

For example

```
In A,(FFh)      reads port FFh and places one
                byte of data into the accumulator

Out A,(FFh)     writes one byte of data from the
                accumulator to the device which
                is addressed by port FFh
```

The forms In r,(C) and Out (C),r allow the flexibility of reading or writing data with any register. Caution, remember that C contains the port address. The Out (C),r instruction does not affect any flags, while the In r,(C) affects all the flags, except Carry, according to data which was read in.

Register C must be loaded with the port address, prior to use, as in these examples:

```
Ld C,FFh        reads port FFh and places data
In B,(C)        in the B register

Ld C,FFh        writes data from the B register
Out B,(C)       to the device addressed by port FFh
```

You will note the I/O instructions assume you are communicating with some device (printer, monitor, disk, etc.) which is 'addressed' by a port number. The port number is selected by the hardware manufacturer, just as Sinclair did in our computers. You can perform I/O operations on all ports however, the results are unpredictable without a device attached. This is due to lack of pull-up resistors on the data lines. Obviously, there will not be any communication if there is no device attached or an incorrect port number is used.

Since we have to contend with devices that are much slower than the CPU, we also have to consider timing. I will not get into this subject very deep, as this type of programming becomes very hardware dependent.

The timing problem is obviously one of slowing down the I/O operations, in an effort to match the device. Let's consider the simple case of reading a switch. We might wish to read the switch once per second, to eliminate multiple switch closures (a good example is in debouncing the keyboard switches).

We can perform this type of delay by looping for a predetermined time period. A simple delay routine that can be used, without destroying any registers is:

```
Delay   Push BC         ;save these registers
        Ld B,xx         ;xx = # of ms to delay
Dly1    Ld C,yy         ;yy = 1 ms delay count
Dly2    Dec C           ;loop for 1 ms
        Jr NZ, Dly2
        DJNZ, Dly1      ;loop for # of ms
        Pop BC          ;retrieve registers
        Ret             ;end delay
```

The value xx is the number of milliseconds to delay and yy is the number of loops needed for a delay of one millisecond. I will not take you through the steps of counting the delay as I wish only to demonstrate the technique. When you are ready to use this routine, you will not need my help with the values xx and yy.

Another method of delay can be used with 'smart' devices, such as a printer. This method uses two separate loops, instead of the nested loops we just looked at. Our example assumes the printer (actually, it's interface) is wired for port 7Fh and it sends a zero byte when ready to accept data.

```
Ready?  Ld C 7Fh        ;get port address
        In A,(C)        ;get ready status from printer
        Jr NZ,Ready?    ;loop unless zero byte received
Print   Ld A,data       ;get data byte to print
        Out C),A        ;send data to printer
        Ret             ;end delay
```

This method has the advantage of not sending any data, unless the device is ready, therefore no data is lost. Can you imagine how this article would look if some characters were lost in transit to the printer? No, that's not what happened, I just write poorly.

There is another solution to this timing problem, which uses hardware. We will not discuss that here, but you should be aware of it.

You also note some I/O instructions on the chart that I have not explained. These perform block I/O operations and will be explained next time, with the rest of the block instructions. They are included here so that it will be clear they are I/O instructions.

By now many of you are undoubtedly trying to write your own MC programs. I wish to give some tips and hints, that will make the process less painful.

First, DO NOT attempt to write a large MC program on the first try. Instead, take the approach we have followed here and write small routines that do a specific job. They can be easily called from Basic and will return to the next Basic line. I would suggest you take a small working Basic subroutine and try converting it to MC. An arithmetic routine is the easiest to convert, as long as it does not contain special functions, such as SQR, COS, etc.

Write your MC in modules (subroutines) that can be easily tested and debugged. This also allows you to develope a library of known, debugged routines that can be used again. Look closely at the routines I have provided in this series. You will note that they are very similiar to each other.

I do not flow-chart and will not describe that to you. There are many good books on the subject. However, before you begin codeing your routine, there are some questions you need to answer or data to collect.

1) Purpose -what do we hope to accomplish?
2) Examples -what happens if? try several tests
3) Inputs -what data does routine need upon entry?
4) Outputs -what data is returned to calling program?

I also strongly encourage you to document you program. All of us have purchased programs that were not user friendly, in spite of it's claims, and in addition, had no documentation. This is deplorable, but the biggest reason for documenting you own programs is for ease of use. I have written code, been interrupted, and when I returned to it a few months later, I could not determine what the code did or why I wrote it that way! Some essentials to proper documentation include:

1)Purpose -if the above questions were answered, they should be included here
2)Registers which ones are used? which ones are destroyed and which are preserved? what should they contain upon entry and exit?
3)Inputs -what data does routine need upon entry?
4)Outputs -what data is returned to calling routine?
5)Routines Callled By -what routines use this one as a subroutine?
6)Routines Called -what routines does this one call?
7)Commented Source -an absolute necessity

This is not the only information that should be in your documentation, but it is enough to make that routine useful to you next time around. Without this information, you will not develope a useful library of routines and will continually need to reinvent the wheel. If you follow these suggestions, you will find MC programming easy (well, almost) and if not, you will soon give up in frustration.

Now, how about a short routine? Let's convert binary numbers to decimal digits for printing.

The easiest way to accomplish this is by repetitively subtracting powers of ten from our binary number and counting the number of times the subtraction is possible better known as division. For the more advanced, try doing this by using the shift and rotate instructions. I am using the subtraction technique, as the code is much easier to follow

```
;Set-up Demonstration
;********************
;
,Inputs: none
;Outputs: print decimal number
,Routines Called. Bn2Dec
;Routines Called By: none
;Purpose: set-up hl for our conversion routine
,
        Org 7530h
Set-up  Ld HL,4000h  ;hl=number to convert
        Call Bn2Dec  ;go convert it
Done    Ret          ;converted and printed
                     ;this is our return to
                     ;basic

,
,our routine really begins here
,
,Convert Binary to Decimal
,*************************
;
;Inputs: HL-Binary Number
;Outputs  decimal number is printed
,Routines Called  Divide
;                 Print
;Routines Called By. Set-up
,Purpose: convert binary number to decimal
;         ascii characters for printing
,
Bn2Dec  Ld BC,D8F0h  ;-10 000
        Call Divide  ;go get 10^4 digit
        Ld BC FC18h  ;-1,000
        Call Divide  ;go get 10^3 digit
        Ld BC,FF9Ch  ;-100
        Call Divide  ;go get 10^2 digit
        Ld BC,FFF6h  ;-10
        Call Divide  ;go get 10^1 digit
        Ld A,L       ;a = 10^0 digit
Exit    Jp Print     ;go print 10^0 digit
,
,
;Divide by 10^x
;**************
;
;Inputs: HL=Binary Number
;        BC=10^x
,Outputs: A=decimal digit to print
,Routines Called: Print
,Routines Called by: Bn2Dec
,Purpose  divide binary number by power
;         of ten to obtain decimal digit
;         by repetitive subtraction
;
Divide  Xor A        ;clear our counter
DvLoop  Add HL,BC    ,perform subtraction
        Inc A        ;count it
        Jr C,DvLoop  ;do again if possible
        Sbc HL,BC    ;otherwise adjust the
        Dec A        ;counters for the extra
                     ;subtraction
        Ret Z        ,division not possible
DvDone  Jp Print     ;go print it
;
```

Chart 1

In A,(n)	Out A,(n)
In r,(C)	Out (C),r
Ini	Outi
Inir	Otir
Ind	Outd
Indr	Otdr

```
,
;Print Ascii Character
,********************
;
;Inputs: A=decimal digit
;Outputs: digit in A is printed
;Routines Called: Rom Print
;Routines Called By  Bn2Dec
;                    Divide
;Purpose: call rom print routine while
;         preserving the registers
,
Print   Push HL      ,save all registers
        Push BC
        Add A,30h    ;2068 only
                     ;make an ascii character
        Add A,1Ch    ;1000 only
                     ;make an ascii character
        Rst 10h      ;go print digit in A
        Pop BC       ;restore registers
        Pop HL
PrDone  Ret          ;digit has been printed
,
End
```

There are several things to note First, since each routine is a separate module that could be called from anywhere, there are a few unneccessary instructions. For instance, the Jp Print is not needed at the DvDone label, as the Divide routine could simply 'fall through' to Print. I used Jp to demonstrate the use of another routine s Ret instruction, in place of a Call Print and the subsequent Ret that would have been needed to end the Divide routine. Assume for a moment that Print does not follow immediately behind Divide. Try to follow the program through and see that the Divide routine uses the Ret instruction from the Print routine to return to the main routine, Bn2Dec

Also, I used the Rst 10h rom print routine for compatibility on both the 1000 and 2068. Use of the rom routines often destroys the registers, therefore they were saved. Even BC, which we could have discarded

The source is written along the quide.ines given above. You should note that the comments do not echo the instructions, except when it serves to clarify I have seen many listings that look like

```
        Ld HL,4000h     ,HL=4000h
```

Obviously not very informative or useful.

Several lessons ago, I made the rather obnoxious claim that all arithmetic could be performed with addition. This routine will perhaps clarify that statement. We needed to divide. We chose to subtract, to achieve this. We chose to add a negative number, in lieu of subtraction. We divided

As a friend of mine says, "Th-Th-Th-That's all folks!", that is, until next issue.

Programming Concepts

by Albert F Rodriguez

Our example program, ZX TIC TAC TOE (the listing appeared in May/June '87 on pages 21 & 22), has the LOAD name "TTT". When storing a program on a cassette, it is better to give it a name, especially when the tape contains other different programs. The procedure is, of course, to use a REM statement with the name of the program within a pair of quotations (see the chapter in the User Manual regarding LOAD/SAVE).

ZX TIC TAC TOE is self-running and this is possible because of line number 5. This line permits the program to begin execution immediately after it is loaded. It also eliminates the danger of the user accidentally erasing any data contained in the variable store by entering RUN. Any program written with a SAVE command, after it is loaded, will begin to execute from the first line after this command.

Lines 7 through 12 are in charge of initializing the character array C$(9). This array is initialized before the beginning of each game with the numbers 1 through 9. These values are what first appear in each square of the game board and permit a player to make a move during a game. The way the loop is structured is interesting from a programming viewpoint. This way of initializing a character array is faster ans consumes less memory, than say, using LET and listing each individual array element in consecutive order.

In lines 16 through 18 the number array N(9) is initialized with zeroes before each new game. This array appears in the driver by the name of COUNTR. This array is used by this routine to store how many times during a game a particular move is made. This is important because it prohibits any one square in the game board from being used more than once during an actual game.

Line 34 initializes the number variable CTR, which is used to keep track of the total number of moves made during a game. When CTR equals nine, it cues the driver called DRW. This routine does what its name says: given 9 consecutive moves and no winner, the game must end in a draw.

Line 35 is really interesting. This line lets the character variable F$, be used further down the program by pre-assigning it at the beginning of each game. F$ in this program acts as a "flag"--it tells the computer that a game has finished in either a win or a draw.

Lines 112 through 120 compose what is actually the main program within my overall program. It has 5 subroutines and two drivers (a clearer example of this will be shown later when we list the program "declarations"). After the last driver at line 119, terminates execution (i.e., when a game being played ends either in a win or a draw), then line 120 permits the start of a new game by letting the program re-execute again from line 7. This way of structuring a program, which involves integrated multi-functional routines, is convenient in that it first allows a programmer to outline the main areas of his/her program, then the rest of the time can be devoted to writing each specific section of the overall work. (I credit this suggestion to Dr. William T. Kraynek, Associate Professor, Mathematical Sciences Department, Florida International University, Tamiami Campus, Miami, Florida.)

From lines 1001 to 1019, two things happen. Subroutine MSG prints a message on the screen for three seconds urging the player to beat the computer. After clearing the screen another set of instructions appear for 12 seconds informing the user how to stop and restart the game. The length of time that each message appears can be reduced by pressing any key, except the space key, while each message is being displayed. These messages are displayed once per each new game.

Lines 2001 through 2012 is the subroutine BRD. This section prints on the screen, once per game, the playing board. The vertical and horizontal lines of the board are made of the characters "I" and "-" respectively. The routine is made of two FOR/NEXT loops and a "counter" by the name of C. The variable C is what informs the computer that three sets of vertical lines and two sets of horizontal lines have been printed and that it may proceed with its next instructions.

Lines 3001 through 3015 is the subroutine PSTN. This name, is actually an abbreviation for the word "position". This routine is continually called upon to display the actual moves that are either made or remain to be made before and during a game. The row coordinates from top to bottom are 2, 7 and 11; the column coordinates from left to right are 2,7 and 13. This routine prints the content of each element in the array C$(9) at pre-determined locations on the game board in row form, beginning with the highest and ending with the lowest row.

Lines 4003 through 4010 is the subroutine INSTRCS, which stands for "instructions". Once per game, this section displays on the screen the name of the game, the year it was copyrighted, its author's name, who get to play with either of the characters (in inverse video) "O" and "X" and who gets the first move. The player who is assigned "O", is the one who always gets to make the first move. Once a game begins, the instructions at the bottom of the screen (which report who moves with what and who makes the first move) are erased.

Lines 5000 through 5017 compose the subroutine RDAPRV. The basic functions of this routine are to read in the move made by a player during a game, then to determine if the move is included within the only acceptable kind of moves that should be made during a game. Only the numbers 1 through 9 are acceptable moves. This routine is what is called a "search" routine. It seeks to match what is entered with what can only be an acceptable move. If the move is acceptable, the game proceeds with the rest of the instructions in the program. If the move entered is unacceptable, then three things happen: an error message is displayed for two seconds, it self-erases and the player is allowed another move. Whether or not a move that is made is acceptable...what is entered by a player is always displayed on the screen to let him/her know exactly what was entered.

The first driver in the program appears in lines 6000 through 6018. Its name is COUNTR. Its function is to NOT allow a player to make the same move, he/she or the computer makes, more than once. It too acts like a search routine by determining which acceptable move was entered, then it stores in the number array N(9) the number of times that move was made. If a move in a game is made twice or more, and error message is displayed for two seconds, self-erases and allows another move by calling subroutine RDAPRV. If the move that was made has not been made before, the program then continues.

We'll wrap up our explanation of the example program next issue.

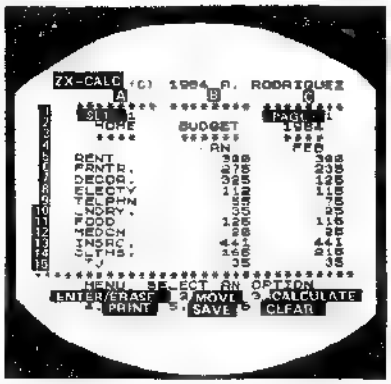
POWERFUL AND INEXPENSIVE BUSINESS SOFTWARE FOR ZX81, T/S1000 and T/S1500 COMPUTERS

ZX-TEXT

A word processor is to a computer user what a typewriter is to a typist except that the former has more advantages than the latter. ZX Text can operate in 16-64K RAM providing from 1300 to 6500 words per document. It features 6 different options: write, read, edit, print, save and clear text. Text is written on a per-line basis with quick speed and with horizontal back space and delete capabilities being available. You can also access the editor directly from write mode and vice-versa. Text can be proof-read on a per-line basis allowing for enough time to determine if any editing is needed. The text editor allows a line of text to be deleted, inserted, replaced and listed for editing. You may also change a word or expression within a line, stop or start text while it is scrolling up the screen, begin reading text from the first line of the file, re-enter write mode from the editor, return to the main-menu or create a window so that you can read-edit two files simultaneously. The print option takes text displayed in 30-column format on the screen and outputs to either the ZX/TS printer (With Memotech's Centronics Parallel Interface 80-column and lower/higher - case output is possible.) Files may be saved on tape cassette with the use of one single command, or by the same token they can be erased from memory / storage so that the full capacity of the program can be used for other purposes such as composing letters, reports, articles, memos, standard forms, instructions, ads, graphs, telephone directory, lists of customers, members, friends etc. Also copies of files are always less expensive and easier to run than using a photocopier. Other advantages are savings in time, paper, ink, correcting mistakes and adding afterthoughts more efficiently than doing them through either handwriting or using a typewriter.

$16.95

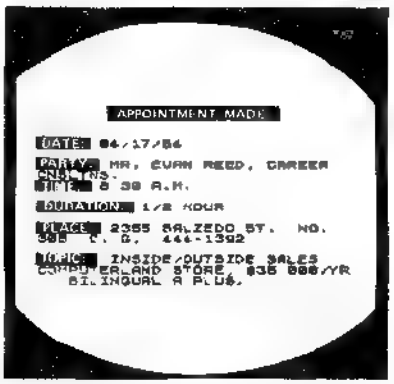
ZX-CALC



An electronic spreadsheet calculator is the fundamental basic tool for summarising, reporting and analyzing in matrix form any accounting, mathematical or scientific manipulation of numbers. ZX-Calc operates in 32-64K RAM and affords a maximum of 3360 characters / spreadsheet. The entire matrix consists of 15 columns (letters A-O) and 30 rows (numbers 1-30) with 8 characters/cell. Unlike other popular ESCs, ZX-Calc uses in calculations and within cells all 14 math functions on the ZX 81/TS1000. It offers a unique "SUM function that totals one or more rows/columns simultaneously. Parenthesis can be used within equations. There is no fixed limit on how many equations may be entered. Formulas may be stored in all 420 cells of the spreadsheet. The display affords 15 rows/colums. Loading of data into more than one cell can occur across/down one or more row/column simultaneously. With vertical windowing you can arrange a set of columns in any order or practice using fixed-variable-alignment display formats. The menu offers 6 options: enter/erase, move, calculate, print, save and clear the spreadsheet. Enter/erase allows the entering, deletion or data alignment within a cell through the use of a mobile cursor. With the move option you may move around the entire spreadsheet to access any row, column or cell. The calculate option allows you to enter labels, values or formulas into a cell or write and enter equations that will act upon the data already within the spreadsheet. You can also enter bar graphs into a cell in this option. Absolute/relative replication down/across a column/row is also allowed by this option. Also this option allows the automatic calculation of the entire spreadsheet with one single command. Print allows you to output to either the ZX/TS printer the entire spreadsheet by column-sets and row-pages through use of the COPY command. The entire spreadsheet may be saved on cassette tape or you may clear all data from it or erase the program from RAM entirely. The most salient advantage provided by an ESC over specifically vertical applications software is that an ESC provides a reusable framework with which you can compose any specific financial mode rather than just be limited to only one statically fixed format for storing, displaying and manipulating numerical data.

$16.95

ZX-CALENDAR



Time management is an important aspect of any serious business and personal agenda. Planning how to spend our time leaves us better prepared before and while we are spending it and we remain better organized after we finish spending it. ZX-Calendar operates in 16-64K RAM affording 25 appointments in 16K, 100 in 32K or 180 in 48K and 64K. Each appointment record holds a maximum of 220 characters. The main menu includes enter, search/check/sort, change, save, clear and print any and all appointments made on a specific date or with any party. Output to either the ZX/TS printer is permissible. This program will permit you to remember to do something or to be somewhere important by cataloging your answers to six questions that you must account for in order not to waste time when it is scarce: when, with whom, at what time, for how long, where and what are you going to discuss and conclude when you get together with someone else? The program lets you permanently organize, record, classify, search, sort, calculate, modify, summarize, obtain a written report and store your answers to the preceding questions so that you will not forget what you decide to do with your time. This program identifies your time according to when you are going to spend it and with whom you are going to share it. Through these forms of labeling appointments you are able to verify or modify how your time is budgeted without wasting ink, paper or more time trying to remember what you said to yourself or what someone else said to you or where you placed certain written messages that you now can't find. With this program you will know where you can find exactly what you need to know about where you want to and have to be, or where you have been, before you get and after you got there. Thus, ZX-Calendar will let you plan your time so that you will never have to worry about what is ahead or what came before, for you will always know by using it, to never be caught astray by any time-frame.

$16.95

$3.00 SHIPPING AND HANDLING/PROGRAM

A.F.R. SOFTWARE - 1605 Pennsylvania Avenue, No. 204 - Miami Beach, Florida 33139

DEALER INQUIRIES WELCOME **(305) 531-6464** FLORIDIANS ADD SALES TAX

L-1 © 1984

THE TS 1500 (ZX81/TS 1000) USER MANUAL

YOUR RAM MEMORY

EARL V DUNNINGTON

In order to become more than a novice programmer, it is essential that you understand completely the structure of your Random Access Memory (RAM) and how it functions. This is also a prerequisite for future articles in this series. Much of the material presented in my articles is written to correct, what in my opinion are misconceptions spread or perpetuated by others. I urge you to try the routines presented as proof of the material. I use a TS1500 and this is the reason the ZX81/TS1000 are parenthesized in the heading. This is also another reason to try out the routines on your own computer.

The structure of a computers memory is known as its Memory Map. The BASIC system of the memory map of your computer extends from the fixed address 16384 up to the variable address of RAMTOP. This BASIC system is divided into various areas for filing different types of information a diagram of these areas with their names can be found at the top of page 154 of the TS1500 and page 128 of the TS1000 User Manuals. It is important to understand that this diagram is drawn as if all of the areas are in use at the same time, which is never the case.

Referring to this diagram, the Lower RAM memory extends from address 16384 upwards in addreases to the top of the Calculator stack. The Upper RAM memory extends downwards from the actual address of RAMTOP to the bottom of the Machine stack. Each area in the Upper and Lower memory, other than the System variables area, expands when in use and contracts when not in use. The Display file when in the expanded mode is also a special case. When any of the areas in the lower RAM expand, they push all of the areas above it upward in memory and the Spare area becomes smaller. This is like an accordian file—when information is withdrawn the areas in the Lower RAM memory contract downwards pulling the areas above it down in memory. This is like a vertical file cabinet, the height of which is the available RAM, with an accordian file folder that works up and down, with the lower end glued to the bottom of the cabinet.

The two areas in the Upper RAM memory act in reverse, pushing down into the spare area when in use and contracting back up when not in use. This is like a two division accordian file, the top of which can be fixed at any position below the top of the vertical file cabinet with the bottom free to move. This allows us to open up a third area for filing data or machine code routines. So we now have a vertical file cabinet with two accordian file folders. When the space between the two is less than 36 addresses, we get an "Out of Memory" error remark.

The names in capital letters below the arrow heads, or arrows as the case may be in your diagram are the names of what are known as the System variables. These consist of two bytes containing the address of the boundaries of the BASIC system areas. They are filed along with many others at fixed addresses in the Systems variables area. A list of the System variables and their addresses starts on page 160 of the TS1500 or page 134 of the TS1000 User Manuals. The BASIC commands to POKE or to PEEK the addresses are given at the top of the page.

Figure No.1 of this article is a diagram of the RAM portion of the memory map immediately after the computer has been turned on and the cursor appears on the screen or after NEW has been entered and the cursor appears. Compare it with the diagram in your manual. I immediately see some mental hands raised! Where is the Program area? Until a program line is typed and entered the Program area does not exist. Where is the Line being typed + work space? It also does not exist until one or more characters are typed into the computer. Where is the USR routines area? It also is non-existant unless actual RAMTOP is lowered in the case of the ZX81/TS1000 with or without the RAM pack or the TS1500 without the RAM pack. Why do you show the Display file as having 26 bytes or 793 bytes when the second paragraph on page 129 of the TS1000 and the fourth paragraph on page 156 of the TS1500 manuals say "When the total amount of memory (according to the system variable RAMTOP) is less than 3.25K, a clear screen—as set up at the start or by CLS, consists of just 25 ENTERs."? In the minimum Display file mode, anytime a character is placed into the file it expands. During initialization a cursor character is placed by the ROM routine into the Display file, expanding it by one byte to 26. In the expanded Display file mode (the address in the System variable RAMTOP is 19712 or greater) the size of the file is 25 ENTERs + 32 times 24 space characters for a total of 793 bytes. As characters are entered into the file they just replace the spaces, having no effect on the size of the file. Under certain conditions such as during a SCROLL, the normal size of the Display file will change.

For those having a ZX81/TS1000 with a RAM pack attached, let's fool the computer into thinking you have less than 3.25 RAM. Turn on the computer. Type in and

Continued Next Page.

A QUESTION FOR TIM STODDARD

Can an external keyboard be wired up to a TS1500 in a similar manner as for a TS1000?

Don Berry
Orlando, Florida

Tim Stoddard replies: To compare keyboard connections, position both the TS1000 and the TS1500 with their non-component sides up and the expansion connector away from you. The larger connector is on the left on both of the computers and is wired the same except for reversing wires 5 & 6 (counting left to right) on the TS1500. The smaller connector is wired backwards in the 1500 compared to the TS1000. Following is a chart which compares the two computers. Hope this information helps.

large connector

TS1000	1	2	3	4	5	6	7	8
TS1500	1	2	3	4	6	5	7	8

small connector

TS1000	1	2	3	4	5
TS1500	5	4	3	2	1

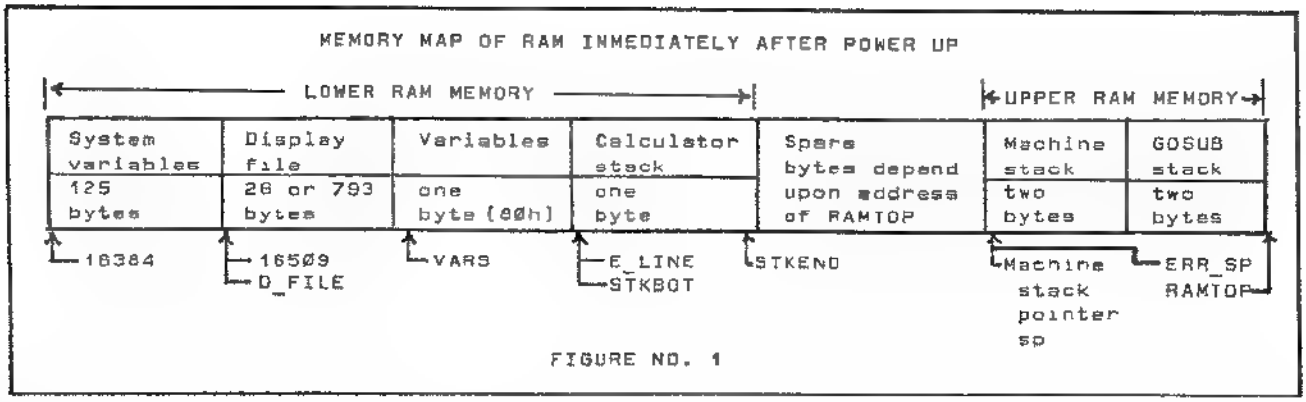


FIGURE NO. 1

ENTER the following direct commands:
POKE 16389,72
CLS
Readers with a TS1500 should do likewise. All others just turn on the computer.

To prove that Figure No.1 is correct, if we PEEK the address contained in the System variable E_LINE and subtract 16384, then the result will be the amount of bytes from 16384 to E_LINE. To do this type in and ENTER the direct command:
PRINT PEEK 16404+256*PEEK 16405-16384
The result should be 151. If you add up the bytes shown on Figure No.1 between address 16384 and E_LINE using the minimum Display file figure, it is 152 bytes. The reason for this is that when you pressed ENTER both the direct command and the cursor character were erased from the Display file prior to the execution of the command. Why did the printing of the value returned and the error remark not upset the value? Because they were printed after the execution of the PEEK and subtract portion of the command.

Let's all fool the computer into thinking we have 3.25K of RAM. Type in and ENTER the following direct command:
POKE 16389,77
CLS
Repeat PEEKing the System variable and subtracting 16384 by typing in and ENTERing the same command used above. This time you should get 919 which agrees with Figure No.1 using the higher value for the Display file.

If we type anything into the computer it will open up and be echoed in the Line being typed + work space area. Because of this we will enter the PEEK commands into a program line. This will not disturb the original displacement between E_LINE and STKEND as after the Line being typed + work space area no longer exist and the Calculator stack has shrunk back to one byte. Type in and ENTER the following program line:
10 PRINT (PEEK 16412+256*PEEK 16413)-(PEEK 16404+256* PEEK 16405)-1
We have to add one because STKEND points to the last byte of the Calculator stack, not to the first byte of the Spare area. Now type RUN and press ENTER. The result should be one byte as shown on Figure No.1.

Turn off your computer and power up so as to have a fresh start. If we take the address contained in the System variable ERR_SP and subtract it from the address contained in the System variable RAMTOP, it will check out the total number of bytes shown for the Upper RAM in Figure No.1. Type in and ENTER the following direct command:
PRINT(PEEK 16388+256*PEEK 16389)-(PEEK 16386+256*PEEK 16387)
The result displayed should be a four.

It is important to understand that any additional memory above the actual address of RAMTOP is not considered part of the BASIC system. This also holds for any additional memory in the 8K area between the ROM and the start of the System variables area at address 16384.

(TO BE CONTINUED IN THE NEXT ISSUE)

T/S 1000/ZX81

PROGRAMMING SRAM HI*RES

PART II

By Fred Nachbaur

In this installment, we'll start taking a closer look at the BLACKJACK listing of Vol. 3, No. 4, to get an idea of what makes it all work. But before we get down to brass tacks, a few other comments are in order.

TS2068 USERS

Are you impressed with the 2068's capabilities, but just a little confused about how to go about using it to the fullest? Well, don't feel too bad; it's actually a pretty complex machine. Even the way its display file is mapped can be pretty confusing. SRAM HI*RES, on the other hand, is simplicity itself. Its display file is arranged very simply as 176 (or 192) lines of 32 bytes each, going from left to right and top to bottom.

So why not dig up that old "doorstopper" and experiment with SRAM HI*RES for a while! Its ease of use and abundance of nifty commands will give you lots to experiment with, and (more importantly) learn from. Once you've mastered SRAM HI*RES, you'll be in a much better position to tackle the intricacies of the 2068.

So here's an "invite" to you 2068ers to join in the fun. Come on in, the water's fine!

WRX16 HI-RES AND INTERNAL 64K

At this point, let's give credit where credit is due. As you may know, SRAM HI*RES Extended BASIC is a collection of new commands that make use of a high-res core routine developed by Mr. Wilf Rigter. This routine is an invention anywhere as significant as the original "cheap video" system employed in the ZX81. Without this core routine, all the neat possibilities that SRAM HI*RES has to offer would be completely useless. With no other hardware requirement than a static RAM in the 8-16K region, this routine is what gives your humble ZX/TS a whole new look, not to mention new worlds of possibility.

The hardware aspect brings me to another point. You may have noticed that the BLACKJACK listing in the last installment was directly followed by Tim Stoddard's 64K BUILT-IN RAM. I checked over the schematic, and guess what? Tim's circuit will work fine with WRX16 and SRAM HI*RES with NO CHANGES! This is because the chips' OE* (output enable) is pulled low if EITHER the RD* line

goes low, OR the REFSH# line goes low. Note that Tim reports compatibility with "THRUST"; guess what THRUST is based on? You guessed it... WRX16.

So, dig up a TS1000 somewhere (I've seen them sell for as little as $15), add about that much more in RAM chips, invest a couple hours of easy hardware hacking, and add SRAM HI*RES. You now have a full 64K computer with high-res capability, at a total cost of under $60. What a deal!

A QUESTION OF SYNTAX

What is meant by "syntax?" No, it's not a payment to your government for vices like tobacco and beer. The dictionary defines it as "The arrangement and interrelationship of words in phrases and sentences." The computer definition could be worded very similarly; "The arrangement and interrelationship of commands and parameters in program statements." For example, there is a definite way of plotting a pixel to the screen in Sinclair BASIC. You have to say, "PLOT x,y". You can't say "x PLOT y" or "PLOT x AND y". If you don't use the correct syntax, the computer won't know what you're talking about; much as if you told someone, "Dog man bites." (Does dog bite man, man bite dog, or does dogman generally bite?)

People who wr te extended language packages (as extended BASIC) for ROM-based computers have a bit of a problem. How do you add new commands to an immutable chunk of silicon? Obviously, we can't (easily) change or expand the ROM itself. Fortunately, there are always other ways of doing things. You are probably used to passing your parameters (like x and y in the PLOT example) using POKE, and executing the command with RAND USR.... Sure, it works. It works quite well. The problems are that this can be confusing (unless you use dedicated variables), and more importantly it's very extravagant with memory. To take an extreme example, let's say you have to pass four parameters (as for a DRAW command). You might have to say something like,

```
1000 POKE 20001,X1
1010 POKE 20002,X2
1020 POKE 20003,Y1
1030 POKE 20004,Y2
1040 RAND USR 22000
```

If you'll do a byte-count of these lines, you'll find that to draw a single line on the screen from (X1,Y1) to (X2,Y2) takes a total of 98 bytes!! It doesn't take long to fill up 16K at that rate!

In comparison, let's look at how SRAM HI*RES would handle the same command:

```
1000 IF USR HR THEN LPRINT DRAW;X1,X2,Y1,Y2
```

Verify that this statement takes only 27 bytes. We can therefore do over 3-1/2 times as much drawing in a given memory space! If we condense the word DRAW to D, then we save three more bytes, without affecting the operation of the command.

Another point has to do with expressions. In Sinclair BASIC, we can say exotic things like, "PLOT 2*ASN (X/2),Y+3/(LN (C/D))". Some extended BASIC systems require you to first assign the value of the expression to a single-letter variable, or POKE the expression into some machine-code variable first. Not SRAM HI*RES! To change the above command to SRAM HI*RES syntax requires only that we precede it with "IF USR HR THEN..." That's all it takes!

Consider the statement "IF USR HR THEN..." as nothing more than a prefix that tells your computer that it is supposed to do something in high-res. Don't worry about "what if not USR HR?" etc. This syntax construct is nothing more than a bridge, or a door, into SRAM HI*RES. There are other subtleties regarding high-res syntax, such as the significance of semicolons and commas in PRINT statements, which we'll get to in the course of this series. For now, simply get comfortable with all those IF USR HR THEN... prefixes; remember that ALL they mean is that we're doing something that we can't do in normal BASIC.

Similarly, the prefix IF USR HR THEN LPRINT... is simply our way of adding new commands with the greatest possible flexibility in using expressions. We call these the "Group 2" commands, as opposed to the "Group 1" commands like IF USR HR THEN CLS, IF USR HR THEN PLOT x,y and so on. Obviously, there are no Sinclair BASIC analogs to commands like SPRITE MOVE, DRAW, DEFINE UDG, etc.

THE AMAZING UDG

What is it that REALLY sets high-res BLACKJACK apart from previous Blackjack games for the ZX81/TS1000? Is it the game itself? No. In fact, the core for this game was originally written on the first 2K TRS-80 pocket computer! It performed flawlessly, if with a few less features. No, the appeal of high res BLACKJACK is in the pictures of the actual cards themselves, right there on your TV screen. It is the GRAPHICS that turns a good but rather ho-hum game into something that will make your friends ooh and aah over your "monster doorstopper".

The key to these pictures is a thing called a "UDG", for "User-Defined Graphics." Though it sounds rather ominous and forbidding, it's really very simple. A UDG is nothing more than a screen character that YOU define. Unlike the 128 fixed characters in your ROM, the 128 UDGs can be defined and changed at will. Every character, UDG or ROM based, consists of eight rows of eight dots, for a total of 64 dots. Each of these dots are either set ("ink") or reset ("paper"). The possibilities are staggering. I once computed that if you printed all the possible UDG's on TS2040 paper, 32 UDG's per line, the length of 2040 paper you'd need would equal the circumference of the orbit of Pluto.

The EASIEST way for a user to define UDGs is by using direct binary, as does the 2068. In other words, "0" represents a reset pixel, "1" represents a set pixel. The only problem with this is that it is tremendously memory-inefficient. For each UDG definition, we would need 64 bytes, plus the "syntax overhead" (punctuation between groups, etc.) Well, so let's use decimal numbers to give the value for each of the eight rows. After all, everyone has ten fingers, right? Unfortunately, 10 is not an integral power of 2; this makes decimal FAR more confusing, in the long run, than other bases like octal or hexadecimal.

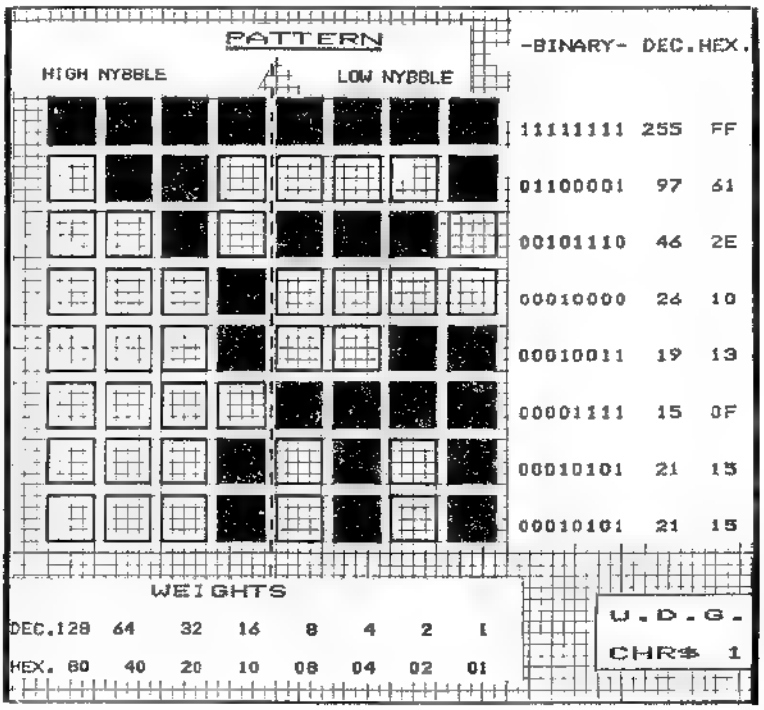


Figure 1

The most memory-efficient approach, while maintaining ease of decoding is (you guessed it) hexadecimal. Time for a philosophical digression. I cringe every time I see an article by some neophyte, arguing something to the effect of, "If God had wanted us to use hexadecimal, He would have given us sixteen fingers," or "if my computer understood hex, I could enter hex numbers as program lines." I cringe not so much because the author is missing the whole point, rather out of embarrassment because I once felt the same way. However, the deeper you delve into the workings of your computer, the more you are forced to realize that

it really IS a "base 2" device (i.e. nothing but ON or OFF). It is very frequently REALLY handy and NECESSARY to be able to quickly convert between a number and its binary equivalent, or vice versa. It is much, much easier to do such conversions using hexadecimal. If you really need the decimal equivalent, look it up in a table or have your computer figure it out using one of the many conversion routines that have been published. A case in point is defining UDG's.

Take a look at Fig. 1. The UDG character depicted there is an actual UDG used in the BLACKJACK program. If you compare it to the screen dumps published in the last installment, you'll see that this is the upper-left corner of the image of the KING card (the left side of the crown at the top of the card). Each row of pixels is "disassembled" in binary, then in decimal, then in hexadecimal. Note that, in the top row, all pixels are set (darkened). Note that each bit (pixel) in this row represents a power of two. The rightmost bit is 2^0=1, the next to the left is 2^1=2, the next is 2^2=4, and so on. So, to figure out the value in decimal, we have to add 1+2+4+8+16+32+64+128 and finally come up with 255. There MUST be an easier way.

Rest assured. There is an easier way. Simply split your image down the middle, as shown by the dotted line in the diagram. So now, each row of eight bits (= 1 byte) is broken into two rows of four bits (= 1 nybble). Now consider the following table of binary-hex equivalents:

```
0000=0 0001=1 0010=2 0011=3
0100=4 0101=5 0110=6 0111=7
1000=8 1001=9 1010=A 1011=B
1100=C 1101=D 1110=E 1111=F
```

Commit this table to memory. Burn it into your mental EPROM. Practise it until you're SURE that 1011 means "B". An easy way to cheat (comparable to counting on your fingers) is to remember that the "weights" (going from left to right) is 8, 4, 2, 1. So "1011" is 8, plus (no fours), plus 2 (9, A) , plus 1 (B). Another way to cheat is to keep the above table handy.

OK, now here's the punch line. To convert a binary number to hex, simply break it into nybbles as described above, then convert each nybble to its equivalent single-digit hex number. The resulting two-digit hex number gives the hex equivalent! For instance, the third line in the diagram is binary 00101110. Breaking this into nybbles we get 0010 (2) and 1110 (E) = 2E. It will take very little practise to get the hang of this; it won't be long at all before you can figure out the hex equivalent of any 8-bit binary number by simple inspection. To figure this same number out in decimal, we would have had to add 32+8+4+2. Don't know about you, but I'm tempted to reach for a calculator when I see something like that.

How does this apply, in practise, to SRAM HI*RES and the BLACKJACK program? Take a look at line 9001. What we're telling the computer here is:

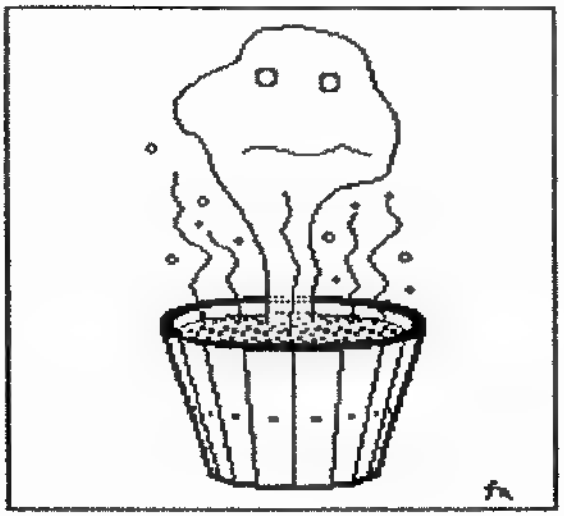
```
IF USR HR THEN ..... [enter SRAM HI*RES]
LPRINT U; ..... [Define UDG]
" ", ..... [starting at space = CHR$ 0]
"00,00,00... etc." ..... [hex numbers for each row]
```

Note that the first eight entries, corresponding to the Sinclair "space" character, are all zero. In other words, our UDG CHR$ 0 is a space just like the "normal" character set. This is where the similarity ends.

Note that, as listed, the "separator" after the eighth hex pair is the graphic symbol on "1". This was simply for my convenience in entering and debugging; in fact, the separators can be anything your heart desires. SRAM HI*RES completely ignores them. The point to remember is, that you can make your UDG definitions as long as you like. The LPRINT UDG; (or LPRINT U;) command keeps processing hex numbers, assigning them in sequential order, as long as they are supplied. So, line 9001 defines the UDGs corresponding to all the characters from CHR$ 0 (space) through CHR$ 9 (graphic on D). Line 9002 starts defining at CHR$ 10 (graphic on S) and continues to CHR$ 19 ("("). And so on, until all 128 possible UDG characters have been defined.

Before we end off, what do we mean by UDG's "corresponding to" the standard Sinclair characters? Very simple. When using the UDG PRINT mode (selected by IF USR HR THEN PRINT;;;;) then anything containing these characters will print as the corresponding UDG instead. For instance, if we said IF USR HR THEN PRINT;;;;CHR$ 1; then the computer would print the UDG shown in the diagram of Figure 1.

VAPORWARE



... From SILICON MOUNTAIN

Now that our SCRAM board is available and off the "vaporware" lists, we felt a need for new entries. We're presently taking orders for the following "new" products which definitely will maybe appear one day.

NERD-WORD

- The word-processor for the compleat idiot. Revolutionary input routine obsoletes the mouse, joystick, light pen, trackball, and even the Keyboard! You don't even have to know the QWERTY Keyboard! Alphabet and control icons continuously scroll across the MCWP (Master Control Panel Window). When the desired character is in the CSF (Cursor Selection Field), press the button on our DBI (Doorbell Button Interface) [only $235.90].

DEBASED

- The file manager of the future! Uses the same input routine as NERD-WORD. Amazing new sort routine; simply print out your files, cut them up, and sort by hand (free tutorial included). Then use our Laser Wand (only $495.87) to read the sorted files back in!

MAIL-PURGE

- Interactively links NERD-WORD and DEBASED. Comes with a pair of scissors and a glue stick for inserting your datafiles into text files. Our patented KAMIKAZE routine eliminates the need for a DELETE function: when the program figures that you've had your files long enough, it assumes that you're tired of them and automatically purges them without saving. Even wipes out the program and gives you an op-art crash-screen!

BUGS-PUNNY

- The ULTIMATE computer game. You are supplied with 16K of pure machine-code, however, the bytes are completely scrambled. Your challenge is to arrange them into a working program. WE don't even know what the final program does! Comes with a packet of instant coffee, a box of double-stuf Oreos, and a pocket bottle of Extra-Strength Excedrin. We'll even PAY YOU $10 if you send us a working copy of the solution!

BUT SERIOUSLY...

We DO have several innovative software titles for your ZX81, TS1000 or TS1500. Send two IRC's for free catalog.

SILICON MOUNTAIN COMPUTERS: C-12, Mtn. Stn. Group Box
Nelson, BC V1L 5R1 CANADA

REMEMBER: In SRAM HI*RES, you can generally use the same syntax constructs as you are used to in Sinclair BASIC. So, any of the following statements would be perfectly valid, and would do the same thing:

```
LET A$=CHR$ 1
IF USR HR THEN PRINT;;;;A$

IF USR HR THEN PRINT;;;;" "

IF USR HR THEN PRINT;;;;CHR$ (2*SIN(PI/6))
```

NEXT TIME...
More on UDG's ... SPRITES ... F-SAVE MAGIC ... MORE SRAM-HI*RES PHILOSOPHY

QL ABACUS/SPREADSHEET "TIPS" PART II

by

Mike de Sosa

One good indicator that QL Abacus (QL SPREADSHEET in the States) is a quality program is the fact that, despite the proliferation of other software for the QL, there is no other QL spreadsheet. It is a first-class professional computer program. The following is intended to go beyond the QL Abacus documentation in the Sinclair QL User Guide in helping you to optimize your use of this excellent program.

TROUBLESOME COMMANDS AND FUNCTIONS

Amend is intended to change the formula in the current (cursor) cell and other cells sharing that formula. (The formula--or data or text--is brought down onto the input line where it may be modified and reentered.) It is more efficient to change data or text in a cell by just reentering it.

Copy is used to copy the contents of one or a range of cells to another place on the spreadsheet. Specify the cell or range of cells to be moved, then the cell or upper left cell of the new range of cells. Formulas included will be adjusted for their new locations.

Design is used to set various parameters of the spreadsheet: the first value shown after each parameter is the default value or option. If ***AUTO-CALCULATE on input*** is set to "NO," the command ***Xecute*** is used to recalculate spreadsheet values. ***CALCULATION order*** ***row or column*** only pertains to the sequential order in which calculations are carried out, which will not usually affect the outcome. ***GAPS between lines on printer*** is a very useful but often overlooked command.

Echo is used to copy the data, formula, or text of a single cell to another cell or a whole range of cells. Formulas are adjusted.

Files offers a submenu of five file handling options. You may import or export data from/to other QL software programs and other QL Abacus programs. Import/Export operations are not complex, but they are a little involved. Consult the ***QL Abacus*** and ***Information*** sections of the QL User Guide, the HELP facility of QL Abacus, and Chapter 12 of my book, if you have it, regarding such operations.

Grid is a key command used to modify the spreadsheet grid in three ways to delete one or more rows or columns, adjusting shared formulas; to insert one or more new rows or columns, adjusting shared formulas but losing data and formulas in rows at bottom or columns at right which are pushed off of the spreadsheet; and to modify

the width of one or more columns

The ***Justify*** and ***Units*** commands work similarly: the former is used to modify the default position of text or numbers in cells and the latter to specify how numbers are displayed within cells. In both cases, the range of cells is specified and whether or not the command applies to already filled cells (selects CELLS) or empty cells within that range (select DEFAULT).

Merge is used to combine data in two or more identically formatted spreadsheets; the merged data may be added to or subtracted from data in filled cells. Formulas are lost. Since you may import and export data between QL Abacus programs, this may prove a more efficient or desirable way to do this.

Order is used to sort rows in ascending order based on values in a specified column. The ordering sequence is empty cells, numeric cells, and text cells (in alphabetic order). In most cases, formulas are lost.

Note the various options available when using the ***Print*** command sequence.

Window is used to split the spreadsheet horizontally or vertically into two movable segments so as to better display cell contents.

Xecute is used to recalculate all formulas in a spreadsheet only if ***AUTO-CALCULATE on input*** is not operative. It is also used to restart a spreadsheet program containing ***askn()*** or ***askt()*** functions--in any event.

An Abacus function converts one or more arguments into a returned value or performs some other function; Abacus functions must be followed by brackets, whether or not it requires arguments.

Askn(text) and ***askt(text)***, "text" being a prompt, request the input of a number or text string, respectively.

Ave(range) returns the average of the numeric cells within the specified range.

Count(range) returns the number of filled data and text cells within the specified range.

Max(range), ***min(range)***, and ***sum(range)*** return the largest, smallest, and sum of all numbers in the specified range of cells, respectively.

Date(n), ***days(text)***, ***month(n)***, and ***time()*** may be used in many imaginative ways in spreadsheets and spreadsheet programs.

If(numeric expression, true, false) returns the "true" argument if the numeric expression evaluates to other than zero or the "false" argument if the numeric expression evaluates to zero. Text and numeric true/false arguments may be mixed.

IRR(range, period) calculates the internal rate of return--the equivalent interest rate--for the numeric data in the specified range (either a row or column segment). This data represents a cash flow for each of several payoffs separated by "period" months--period is the number of months, that is, a number. The example in the User Guide is adequate, if you need more explanation.

Lookup(range,offset,value) refers to an established reference table in the spreadsheet, returning a preset value corresponding to the argument ***value***. The function requires three arguments: ***range*** is the row or column segment containing the entering argument, ***offset*** is the number of rows or columns separating the ***range*** and the row or column segment containing the returned preset (output) values, and ***value*** is the entering argument. Entering and output values must be numeric, the former in ascending value. The example in the Abacus HELP facility is adequate if further explanation is needed.

NPV(range,percent,period) returns the ***Net Present Value*** of cash deposits (the amount of money which must be invested now to produce a specified cash flow at some futuredate, assuming a given interest rate). Three arguments are required: ***range*** is a row or column segment containing cash deposits made at equal intervals of time, the ***period***; ***percent*** is the assumed interest rate. Once again, the example in the User Guide is adequate for a fuller explanation.

THE BOTTOM LINE

QL Abacus is a powerful manipulator of both text and numeric data which will prove highly useful to you in almost any pursuit. You may find it easier to lay out spreadsheets in the following sequence: headings and labels, first; formulas; data; and, finally, editing the format.

TIP: Setting the QL clock is important to QL Abacus (and other programs). Amend your QL Abacus "boot" program to include the following:

```
5 AT 6,5: INPUT "Set clock: YYYY,MM,DD,HH,
  MM,SS"\y\m\d\h\n\s
6 SDATE y,m,d,h,n,s
```

NEW QL PRODUCT: SPECIAL EDITION EDITOR

Digital Precision's ***The Editor***, a text editor program, was subtitled--somewhat prematurely many felt--as "CHUCK QUILL OUT!" It was not a WYSIWYG ("what you see is what you get") editor and had no printer driver. DP's improved text editor, ***Special Edition Editor***, for use on expanded QL's, is still not a WYSIWYG--why I don't know--but it is a most comprehensive text editor, much more flexible than QL Quill, and for those who are sufficiently capable and wish to take the trouble of learning a rather complex system, this may be your cup of tea.

Special Edition Editor should be especially useful to those who must frequently publish sizable documents such as Time Designs Magazine. This is especially true considering that it is designed to be compatible with DP's new ***Desktop Publisher***--said to be good on graphics but less so on text.

Special Edition Editor is far too complex to describe properly in a short article. It now has a most comprehensive printer driver. Suffice it to say that most of the shortcomings in ***The Editor*** have been eliminated, that it has dozens of additional features, and that it will do just about anything that you have in mind. For expanded QL's only! About $50 from Sharp's.

But I still wouldn't chuck Quill out. For all of Quill's faults, WYSIWYG is the only way to fly.

NEXT TIME: QL Easel/Business Graphics "Tips," plus more new QL products information.

SIZZLIN' SUMMER SPECIALS

HERE IS JUST A SAMPLE OF THE GREAT SUMMER BUYS WE HAVE TO OFFER!
(ALL SALE ITEMS LIMITED TO STOCK ON HAND)

****

1000/ZX81

RMG#422	FORTY-NINER (HI-REZ GAME)	REG.:$ 9.95	SALE $ 5.00
RMG#421	ZXTRICATOR (HI-REZ GAME)	REG.:$ 9.95	SALE $ 5.00
RMG#409	ROCKETMAN (HI-REZ GAME)	REG.:$ 9.95	SALE $ 5.00
RMG#105	T/S 1016 16K RAMPAK (NEW)	REG.:$13.95	SALE $ 7.50
RMG#583	TS1000 IDEA BOOK (NEW)	REG.:$ 4.95	SALE $ 1.50

==========

2068

RMG#342	2068 KEYTOPS (PEEL & STICK)	REG.:$ 2.00	SALE $ 1.25
RMG#613	NOVELSOFT-ZXPERT	REG.:$19.95	SALE $16.00
RMG#240	NOVELSOFT-TIMACHINE COMPILER	REG.:$19.95	SALE $16.00
RMG#556	ARROW-MUSIC DESIGN 2000	REG.:$19.95	SALE $14.95
RMG#568	ARROW-SCREENDESIGN	REG.:$14.95	SALE $11.95
RMG#519	2068 ROM MANUSCRIPT BOOK	REG.:$16.95	SALE $13.50
RMG#522	INTRO TO 2068 MACHINE LANGUAGE	REG.:$16.95	SALE $13.50

==========

QL

RMG#280	QL ASSEMBLER PGM	REG.:$34.95	SALE $25.00
RMG#309	QL TRIVIA QLUE	REG.:$19.95	SALE $16.50
RMG#364	WD UTILITY PACKAGE	REG.:$19.95	SALE $10.00
RMG#365	WD PROJECTOR PGM	REG.:$14.95	SALE $ 7.50
RMG#288	QL DECISION MAKER PACKAGE	REG.:$34.95	SALE $25.00
RMG#260	PC-IMPORT	REG.:$39.95	SALE $27.50
RMG#281	QL MONITOR	REG.:$24.95	SALE $19.95
RMG#391	QL SERIES BOOK MC PROGRAMMING	REG.:$ 9.95	SALE $ 8.00
RMG#395	QSPELL	REG.:$29.95	SALE $19.95
RMG#419	POINT BLANK-KARATE	REG.:$24.95	SALE $17.40
RMG#289	EMPTY MDV WALLET (HOLDS 4)	REG.:$ 1.50	SALE $ 1.00

****

IF YOU WOULD LIKE A COPY OF OUR COMPLETE TS CATALOG,
PLEASE SEND $3.00 TO THE ADDRESS BELOW:

RMG ENTERPRISES

1419 1/2 7TH STREET
OREGON CITY, OR 97045
503/655-7484

PLEASE: ADD $2/ITEM SH. TO MAX OF $6 TOTAL.
FOR ORDERS OUTSIDE THE CONTINENTAL US-ADD 5% FOR ADDED SHIPPING.
FOR CREDIT CARD ORDERS-ADD SURCHARGE OF 5% OF TOTAL ORDER.
BE SURE TO INCLUDE CARD # AND EXPIRATION DATE.
PLEASE INCLUDE TELEPHONE # WITH ORDER.

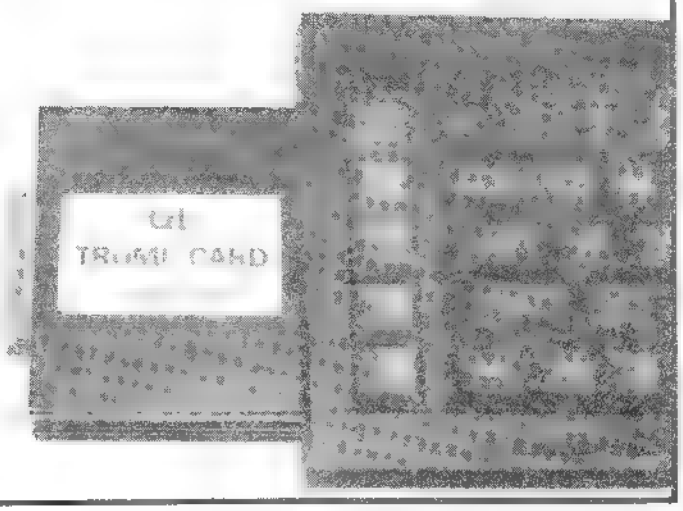
QL Support

Have you heard the latest? A rumor has been circulating in Great Britain for a couple of months now, that Amstrad is rethinking their position on the QL. Amstrad executives have been keeping an eye on how the QL aftermarket support has been developing. Also, it has been common knowledge that Amstrad would like to market a 32-bit machine. Using existing QL technology which they obtained in a major buy-out of Sinclair last year, would save the company thousands of dollars. Having an established support base of software and add-ons is another big plus. Some improvements including the replacement of the microdrives with a single built in disk drive, have been discussed.

Digital Precision has just released [illegible] PUBLISHER software package. It is [illegible] professional program with text editing features [illegible] many similar programs available for the [illegible] and IBM PCs. DESKTOP PUBLISHER is written in machine code and requires a minimum of 256K RAM. Several U.S. dealers have the program available for about $100.

Another Desktop program, FRONT PAGE, has been upgraded for use with additional RAM expansion cards. The new version which incorporates extra fonts and other features, is called FRONT PAGE EXTRA.

The TRUMP CARD by Miracle Systems packs a lot of punch in one circuit card that fits into the QL expansion slot. When installed, it provides a full 896K RAM, a disk interface (up to two drives), RAMdisk software, a print spooler, and Tony Tebby's TOOLKIT II software on an EPROM. Most U.S. QL dealers have the TRUMP CARD in stock for around $300. Miracle Systems is also working on a MIDI Interface which will be ready sometime this fall.



Qool QL

M. VINCENT LYON

Some of you who have never encountered heating problems with your QL before, might discover that about August 15, or the hottest day of the year, that your QL will suddenly blank out. Actually, it is not quite so "suddenly". There are usually small warning signals like a ripple on the screen or the colors may start to "swim". Unless you don't mind losing your data, once you observe these symptoms, it is best to quickly save your data, and turn the QL off for an hour or two.

If you take some of the precautions in this article to heart, there is a good chance that you will never have to worry about the heat problem. Indeed, if you are running an unexpanded QL, without disk drives, or additional memory, heat build up may be so minimal that you will never encounter any problem at all. However, add a disk drive, and you may run into problems.

Even if you do not currently have a heat problem, remember that any heat, over time, is the number one killer of all electronics. Static zaps a few, but heat kills all chips sooner or later. Keeping your QL cool can expand its life.

Rather than starting with the most complex fixes first, I think it wise to attempt to correct the problem with the smallest fix first. If that fails then move on to more elaborate repairs. So lets begin "outside" the QL.

The first thing to do is, if your QL is not plugged into a quality surge protector, PLUG IT INTO ONE NOW! When doing so, make sure that the power cable running from the surge protector to the QL is not on any carpeting, nor is close to your monitor, or any potential source of static. As an additional benefit, should the power company decide to mess up your power, as they did mine, your QL should be protected. (My power company sent surges through my house that blew over 15 of my appliances including the refrigerator, intercomms, radios, a VCR, the furnace and other goodies. But, although my surge protector was burned, and no longer functional, my QL and disk drives were untouched.)

Secondly, get to know the ventilation system on your QL. If you look at the right side, just below the microdrive ports, you will notice a small line of openings. Beside being the speaker port, these openings permit the inflow of air that is supposed to flow over the heatsink, and exit on top at the rear. Noting that a convection flow of this size has, at best minimal cooling capacity, NEVER PUT ANYTHING ON THE RIGHT TOP OF THE QL. In a few tests I have made, one single sheet of #20 Bond placed over the microdrives, raised the internal temperature 18 degrees. It not only insulated the top of the case, it prevented the proper convection flow.

If you have done the first two things, and still encounter heat problems, there is a good possibility that the problem can still be solved for about $20.00, without opening the QL case. Run down to Radio Shack, buy a 3" brushless 9VDC fan (part #273-243). They list for $14.95, but have been on sale recently for $11.88. This fan operates on 7 to 13.8 VDC and supplies .27 CFM airflow. Then add a 9 volt power supply (part #277-1026) for about $4.95, and any switch you wish (less than $1.00), and you've got a system that should solve all your problems.

Hook the fan, through the switch to the power supply. Place it on your desk behind the QL, aimed at the right side heat vent. Before you turn it on, feel the cover over the microdrives. Turn on your fan and wait about five minutes and feel the microdrive cover again. It should now be cool to the touch.

While we are forcing the air in a reverse pattern to the normal convection designed with the QL, the flow is sufficient to really cool the QL, and possibly you as well. Remember that if you aim the fan at the front of the QL, you may be forcing dust into the microdrive ports, which could create more problems than overheating.

I put one of these fans on line and put it through a few tests. With the computer on, but idle, the internal temperature in the microdrive port was 104 degrees. After the fan had been running for five minutes, the temperature had dropped to 68 degrees. I then wrote a program to format a microdrive 15 times, and measured the temperature during the format (drives running create heat). On the 15th format, the internal temperature was still 73 degrees.

I have been advised that RMG Enterprises of Oregon City, OR is in the design phase of a cooling fan system designed specifically for the QL, and should run from the QL's internal power. So, in the future, there may be a system more efficient than this simple one. In the meantime, this seems to be all that is available.

So far, we have found some simple ways to keep a QOOL QL without resorting to opening the case and fiddling around with the insides. If all these have failed, and ONLY IF ALL THESE HAVE FAILED, then we are forced to open up the QL and begin to tackle the problem other ways.

The QL uses a one amp 7805 voltage regulator (you will see it attached to the heat sink behind the microdrives). Since the voltage regularo is heat protected (if it overheats, it shuts down), this, while maybe not the cause of the problem, is certainly the heart of the problem.

The important thing to remember is that simple metal to metal contact is just not an efficient heat transfer system. Metal must be somehow "bonded" to the heatsink if heat is to transfer properly. At assembly, many QLs suffered the same fates. The heatsink compound used was of a type that dries and ages with heat, and in adequate amounts of the compound were used. As a consequence, the unfortunate voltage regulator is unable to transfer its heat to the heatsink.

Fortunately, for us, the fix is simple. Remove the heatsink from the voltage regulator. Scrape off any old heatsink compound, and clean both surfaces as much as posssible. Put a generous coat of any silicone based (it doesn t dry or crack) heatsink compound on the voltage regulator where it contacts the heatsink and reconnect the two. This increase of thermal transfer will permit the heat to flow to the fins of the heatsink, where your fan can efficiently carry the heat away.

Now, even if you're running 640K and three disk drives, you should have no heating problem. But, if you continue to encounter problems, there is an additional hardware fix published previously in QUANTUM LEVELS (Vol 1, No 2) by Tom Woods. I don't recommend that you attempt this fix unless all else has failed. It involves installing an additional 7805 voltage regulator, a capacitor and a diode to attempt solve the problem. The single case that I know of that tried this method was unsuccessul, probably because the simpler fixes, like a power surge protector were not attempted first.

If you wish further information on the final hardware fix, I suggest you contact Tom Woods at Quantum Levels, PO Box 64, Jefferson, NH 03583.

Common sense however, demands that I state, if you have done all outlined procedure, short of the hardware fix, and you continue to have heat problem, you would be well advised to contact A+ Computer Response and get a replacement Q L. If the fan and heatsink compound don't work, you may have serious problem. Historically, A+ has not been really prompt in replacing defective QL's (eight weeks is about standard), but right now, we users seem stuck with that situation.

It may be possible that your QL dealer can provide a temporary "back-up" QL for your use while yours is at A+ for evaluation.

As a final thought, (seems like I'm talking to my kid, now) I must stress the importance of cleanlinesss. Dust blocked vents cannot do their job properly. It may appear insignificant, but dust does inhibit air flow essential to cooling. It also has great insulating properties.

Here's hoping your QL is as QOOL as mine (knock on wood).

Time Designs Tests - - -

by

Mike de Sosa

This article inaugurates a new series in TDM: the most comprehensive report of the testing of new hardware and software for the Sinclair QL yet done in the states.

Each item reviewed will be awarded from one to five stars, depending on its degree of usefulness and excellence, and "turkeys" --those which I judge to be an affront to the consumer--may be given a blivet (0).

*THE NEW TASKMASTER * * * * **

Sector Software's "latest" version of their five-star multitasking software, TASKMASTER, is profoundly different from earlier versions. As many as eight QL software programs may be multitasked at the same time: the preset list of QL software programs to be executed is QUILL, ABACUS, EASEL, ARCHIVE, ABACUS, and ABACUS. Only one program of each type is actually loaded, but each program executed has its own reserved data/manipulation area in RAM. You may choose your own set of QL (Psion) software or other programs and dedicate an appropriate portion of RAM to each. A full set of programs and utilities loads from disk in about 20 seconds!

One major improvement in the new version, and one which should be emulated by other vendors, is that the original master program disk or cartridge need not be used to boot TASKMASTER--a great leap forward!

Four new functions have been added to the TASKMASTER Calculator module (powers of numbers, percentages, etc.) and the final result may be sent to a QL software program.

TASKMASTER has added a NotePad utility which is called by keying ALT F4 (which also calls the COMMAND FILES module). You may Load, Edit (write), Save, Send (to a software program), Zap, or Print a note.

Perhaps the most far-reaching addition to TASKMASTER is its Command module which permits command files consisting of up to 2000 keypresses to be "learned," saved, and later accessed. When accessed, the keypresses will be duplicated.

Two new SuperBASIC keywords, SCRON and SCROFF, and a user definable print buffer.

Be sure to order version 2.35 with serial number greater than 4000! About $36.

*SPELLBOUND * * * **

SPELLBOUND is Sector Software's new spelling checker--and more! SPELLBOUND, designed for use with QL Quill or Digital Precision's EDITOR and SPECIAL EDITOR on an expanded QL, has five levels of operation ranging as the documentation states from the "totally unobtrusive to absolutely insistent." (A star was withheld because SPELLBOUND cannot be used to check a completed document, but I am told a patch will soon be published in Sinclair QL World which will permit this.) SPELLBOUND utilizes a 30,000 word dictionary to check your manuscript as each letter is typed. (This does not in any way interfere with your typing.) Additional words may be added up to the memory limits of your machine, and a list of correctly spelled words displayed on which the correctly spelled word may be indicated and sent to your document.

An audible and visual signal is given when a misspelled word is typed. About $45.

*TOUCH TYPIST * * * **

David Batty's (Sector Software) comprehensive typing tutor is the best instructional typing software for the QL. TOUCH TYPIST has a 1200 word vocabulary, a 200 lesson format, an excellent tutorial, and an on-screen keyboard. Lessons are easily customized to fit any style of teaching. It even has a graph utility which reflects your (or each member of your class's) highest speed obtained on each of the lessons.

I didn't give TOUCH TYPIST five stars because it requires the original program cartridge to be inserted in Microdrive 1, which presents problems when booting from a disk backup--no program that utilizes such copy protection will ever get five stars from me. A truly excellent version of one of the most important instructional software programs. A bargain at about $20.

Freddy Vaccha at Digital Precision has come out with a whole new and revised family of QL software: such things as TURBO (a SuperBASIC compiler even more powerful than DP's SUPERCHARGE); DESKTOP PUBLISHER, perhaps the Rolls-Royce of such software for the QL; and revised versions of EYE-Q, THE EDITOR, and SPECIAL EDITOR (for the expanded QL). I have tested the latest EYE-Q and SPECIAL EDITOR (dozens of new features, including a smart printer driver); both are excellent.

the CLASSIFIED

FREE ADS FOR SUBSCRIBERS

FOR SALE: TS2068 and Aerco printer interface; Flight Simulator & many other cassettes...$100. TS1000, modem, printer, terminal I/F for HAM Rtty, CW, Wepax...for $100. Call: Donn KAOSHOH (314) 832-8617 or write: 5814 Minnesota, St. Louis, MO 631111.

FOR SALE: TIMEX 1000 Software. Flight Simulator, The Gambler, Supermaze, Cube Game, Checkbook Manager. Must have 16K RAM. Best Offer. Tim Kessler, 29 Wine St., Uniontown, PA 15401.

FOR SALE: TS2068 Computer with 3 different manuals, 2040 Printer, Westridge Modem with 2 programs, and Panasonic Tape Recorder. Have 8 programs. 2068 also has Spectrum ROMSWITCH and RGB Cable. All in like new condition. R.L. Sutton, PO Box 280, Bronson, FL 32621 ($200 for all).

MONEY MACHINE $12, Diamond Mike $15. 2068 Software on Aerco disk or tape. New/used hard/soft-ware for 1000 & 2068. 22-cent stamp for catalog. Chia-Chi Chao, 73 Sullivan Dr., Moraga, CA 94556.

WANTED: Three DAMCO RAINBOW INTERFACES (Spectrum Emulators) for TS2068. Leslie E. Kulberg, Rt 2, Fayetteville, TN 37334, telephone (615) 433-1050.

FOR SALE: ALPHACOM 32 Printer, w/ interface & power supply $30. Textwriter 64, Artworx V1.1, Zeus Assembler, Ckt. Bd. Scramble (all for 2068) $40 for all, or trade for full Aerco W/P. W. Flower, 1804 Vincennes St. #22, Northridge, CA 91325.

THE WILD WEST ADVENTURE for 16K ZX81 or TS1000/1500. Machine code and BASIC. Cassette with full printed instructions: $10.00. Larry Dietrich, P.O. Box 13, Blanca, CO 81123.

WANTED-MICROPROLOG, complete as noted in the JAN/FEB 1987 TIME DESIGNS. Send postpaid price to R. Steensen, 1010-H2 Green Pine, West Palm Beach, FL 33409.

DESKTOP PUBLISHER for 2068. Two Versions: Vers 1 works on all dot matrix printers. Vers 2 works on Olivetti 2300 printer. $19.95 & $24.95 (+ 1.25 postage). Send SASE for samples. Charles Stelding 1415 S. Baxter, Tyler, TX 75701.

WANTED: TS-2050 Modem manual. Nick Oshana, 187 Morningside Dr. East, Bristol, CT 06010.

2068 PLOTTING w/Commodore 1520 4-color HI-RES printer/plotter! The I/F hard/software is available from John McMichael, 1710 Palmer Drive, Laramie, WY 82070. Send SASE for complete info & sample plot.

FOR SALE: TS1000 in metal cabinet with keyboard, Westridge modem, Aerco C.P. interface, Memotech 64K RAM. Send for complete hardware and Software list to: Richard Beier, 1 Darwin Dr., N. Merrick, NY 11566.

WANTED: Low Price TS1000 computer, 2040 printer power supply, and modem hardware & software and related items. Call or write: Brian Cornelius, 221 Walnut Lane, Apple Valley, MN 55124, (612) 932-4430.

Do you have some equipment or a program that you would like to sell? Looking for something hard to find? Place an ad in **THE CLASSIFIEDS!** Subscribers can place one free personal ad in each issue. Ad size is 32 Col. wide (like 2040 paper) and maximum of six lines. For additional lines - $3 each. **NON-SUBSCRIBERS** and **DEALERS:** $4 a line. **DEADLINE FOR ALL CLASSIFIED ADS:** Two weeks before publication date. Mail your ad to: **TIME DESIGNS MAGAZINE, The Classifieds Dept., 29722 Hult Rd., Colton, Oregon 97017.**

PIXEL PRINT Desktop Publisher
BY LEMKE SOFTWARE DEVELOPMENT

DESKTOP PUBLISHING: THE ABILITY TO PRODUCE NEWSLETTERS AND OTHER DOCUMENTS OF TEXT AND GRAPHICS HAS JUST BEEN MADE AVAILABLE TO TS 2068 OWNERS...

PIXEL PRINT NEW PRINT SENSATION!

Lemke Software Development of Wichita, Ks. has just added the PIXEL PRINT Desktop Publisher to their program line.

PIXEL PRINT enables TS 2068 users to design and print their customized NEWSLETTERS and other documents using a wide variety of FONT styles and sizes, pictures (graphics), and layouts.

FONTS include: TS 2068 BOLD, MODERN, *ITALICS*, CHANCERY. PLUS, you have the ability to LOAD and use other custom made FONTS.

GRAPHICS (pictures) can be imported from any TS 2068 program that uses the SCREEN$ protocol. (The PIXEL SKETCH and GRAPHICS EDITOR v2.0 is recommended!)

LAYOUTS include a TWIN column format common in NEWSLETTERS and MAGAZINES, or a single L-A-R-G-E column for BULLETIN BOARDS and other PUBLIC NOTICES, or use a combination of both.

PRINT to most IBM/EPSON/STAR compatible DOT MATRIX printers, (PROWRITER too!). Uses the AERCO TASMAN, A&J Microdrive CPIs.

THIS AD WAS DESIGNED WITH THE PIXEL PRINT DESKTOP PUBLISHER

Get YOUR PIXEL PRINT now from:

LEMKE SOFTWARE DEVELOPMENT
2144 WHITE OAK
WICHITA, KS. 67207
PIXEL PRINT JUST $19.95 PPD.

Lemke Software Development

S D Lemke
2144 White Oak
Wichita, Ks.
67207

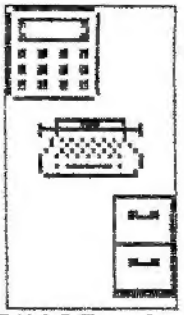
(Wichita, Ks) - Desktop Clip Art and Font support announced!

Lemke Software and Mountaineer Software have joined forces to provide the PIXEL PRINT user the best utility support possible with the PIXEL PRINT ICON PACKAGE of Desktop Clip Art (pictures), and the PIXEL PRINT FONT PACKAGE a collection of 6 NEW fonts.

The PIXEL PRINT ICON PACKAGE is a collection of 3 programs and over 100 ICONS!

#1) ICON LIBRARY MANAGER allows you to browse a library of 115 ICONS, adjust its size, location and place a frame around it.

#2) ICON DESIGNER PROGRAM gives you the ability to make your own (custom) ICONS or modify existing ones This is a high quality design program tailored for ICONS. (Joystick based, fast, fun!)



#3) COLOSSUS BANNER Conversion, converts these ICONS to the low-resolution format for COLOSSUS BANNERS.

The PIXEL PRINT FONT PACKAGE is a collection of 2 programs and 6 new FONTS. NOTE: these FONTS are compatible with the PIXEL SKETCH and GRAPHICS EDITOR v2.0 and the COLOSSUS GRAPHIC BANNER programs as well as the Desktop Publisher

#1) FONT DESIGNER allows you to design your own FONTS, or modify existing FONTS! It is also a FONT librarian allowing you to browse a library of up to 20 FONTS.

#2) FULL SIZE PRINTER FONT DESIGNER and DOWNLOADER program. This utility is an all BASIC program written to design and to download FONTS to a STAR (SG-10) printer. It is only intended as an example for others who wish to explore this aspect of FONTS.

The 6 new FONTS are:

HEADLINE, IBM, MICR,
Adventure, Outline, SXS

(Most of this ad was written in the BOLD-IBM FONT.)

Get YOUR PIXEL PRINT ICON and FONT utility packages from LEMKE SOFTWARE, just $19.95 each (ppd)

WEYMIL presents...

A small collection of truly innovative products for Sinclair computers

Introducing the Delta Device

The Non-Volatile Memory System for the TS 1000

The DELTA DEVICE is the newest addition to our family of TS 1000 products. It is a complete non-volatile memory system (NVMS) designed by Mr. Wilf Rigter. The DELTA DEVICE is a full 32K non-volatile memory board featuring the Rigter Operating System (ROS). We feel the features of this product make it a significant advance in TS 1000 technology.

THE HARDWARE

The board measures 3½" x 3". It uses only two chips. It is supplied fully assembled. There is a built-in write protect switch to avoid accidental erasure of data. The 32K is divided in four 8K blocks which are individually controlled via DIP switch for mapping to various memory map locations. A long-life battery preserves memory contents. The circuitry has been designed to greatly reduce the risk of data loss caused by removing the unit from the computer. This makes the system very transportable. The board utilizes a standard feed-through connector. Memory control is extended to both 16K and 32K rampacks. Bank switch applications are supported. There is built in hardware compatibility with THRUST and other hi-res programs. The hardware has four times the capacity and more flexibility than other similar memory enhancement products.

RIGTER OPERATING SYSTEM

The NVMS concept is completed by the RIGTER OPERATING SYSTEM (ROS). This is a full powered file handling system. Utilizing less than 600 bytes, ROS has these features. DIRECTORY displays all programs stored in memory by name and length in bytes. You have 44 entries per directory. SAVE transfers BASIC programs, variables, and machine code to storage. It works in conjunction with the NAME / RENAME features. LOAD transfers those programs to their normal RUN areas. MERGE allows the joining of two BASIC programs or varible files. EXIT allows you to quit ROS to an auto-run program or the command line. ERASE deletes programs from system memory and automatically moves other programs to fill the space left behind. This eliminates blank areas of memory between files. File selection is accomplished using cursor movement. ROS utilizes terminate-and-stay-resident protocols. It is called from the command line by simply pressing REM followed by ENTER.

There are further enhancements to the ROS core. CLEAR DIRECTORY allows initialization of selected sections of memory while leaving others intact. RENUMBER is a natural companion to the MERGE feature. UNMERGE deletes blocks of BASIC programs. The comprehensive 15 page manual includes instructions for adding more directories and "hot keys."

APPLICATIONS

The DELTA DEVICE has many possible applications. A user can have programs such as NOVA, THRUST, MINI-XMOD, KRUNCHER, and a word processor instantly available with plenty of room left over. Other TS 1000 NVM devices can't equal that capacity. You can easily enhance the ROM by mapping a section of the NVMS into the ROM area. You can now design a customized prompt, develop hi-res graphics, or install a high speed tape loader all accessible using the regular command keys. Programmers will appreciate the ability to have more than one operating system available on the same machine. Imagine having one machine with FORTH, PASCAL, and the standard operating system instantly available.

We are sure that you will agree that Wilf Rigter's DELTA DEVICE is an incredible amount of power in a tiny package.

THE DELTA DEVICE NVMS for the TS 1000	**$75.00**
Shipping and handling	$5.00

WEYMIL CORPORATION

Box 5904

BELLINGHAM, WA 98227-5904